$\mathbf{E}$asy and $\mathbf{A}$ccurate $\mathbf{Z}$(photometric redshifts) from $\mathbf{Y}$ale

Gabriel Brammer, Pieter van Dokkum, Paolo Coppi
Department of Astronomy, Yale University

(Incomplete) Draft Version: May 14, 2008

Contents

1 Introduction

EAZY is a public photometric redshift (phot-z) code designed to incorporate the most recent phot-z innovations into a flexible yet easy-to-use interface (à la HyperZ - Bolzonella et al. [2000]) with default settings that will produce high-quality results without significant optimization by the user. The default settings have been optimized to work with multi-wavelength photometric data alone and therefore EAZY does not require a large "calibration set" of sources with measured spectroscopic redshifts to obtain optimal phot-zs. Important features of the code are summarized below:

- Fitting is done in linear units to naturally handle negative fluxes
- Fit one, two, or many combinations of templates simultaneously
- Incorporation of a redshift prior, determined from the "Millenium lightcones"
- Template set optimized on the lightcone photometry
- Template error function
- Redshift quality parameter, "Q_z"

In the text below, parameters that EAZY uses are printed in SMALL CAPS, contents of ASCII files are printed in `typewriter` font, UNIX terminal commands are shown as, e.g., `$ ls`, and literal filenames are printed in *italics*.

This document is intended as a practical guide to setting up and using the EAZY software package. For a higher level discussion of the algorithms summarized below and for extensive tests of the code on publicly-available photometric datasets, see Brammer, van Dokkum & Coppi [2008].

2 Installation

EAZY does not require a particular directory structure or any environment variables to be set. The example below shows the structure of the program files as they're distributed, but you can put things anywhere you want making sure to change the appropriate file paths in *zphot.param*. To get started, download the tarfile of the distribution from the website,

`http://www.astro.yale.edu/eazy/`

and unpack it:

```
$ tar xzvf eazy-1.0.tar.gz
```

Now compile the code and run it:

```
$ cd eazy-1.0/src
$ make
$ cd ../inputs
$ ../src/eazy
```

And you're done. Well, not quite yet. EAZY requires a file, *zphot.param*, in the working directory that contains a list of parameters for the run. In this case EAZY didn't find the file, so it created a file for you, *zphot.param.default*, containing all of the default parameters. We'll use the defaults for this first run on the example catalog, so do

```
$ cp zphot.param.default zphot.param
$ ../src/eazy
```

Now you're done and you have computed phot-zs for the HDF-N catalog of Fernández-Soto et al. [1999] with $\sigma_z/(1+z_{\rm spec}) \sim 0.04$. EAZY dumps a lot of information to the terminal so you can see what it's up to. For some UNIX terminals, printing to the screen can take up a significant fraction of the CPU time, so you might want to pipe the output to a file (where you can then actually read it):

```
$ ../src/eazy > logfile
```

3 Parameters and input files

3.1 *zphot.param*

This is the main ASCII file that contains the parameters needed to run EAZYṪhe code is not too picky about the formatting of this file; it just looks for lines formatted like

PARAMETER ***value*** [# OPTIONAL COMMENT]

and assigns the value to the parameter if the parameter is one that EAZY recognizes. If the parameter is not recognized, the line is ignored. If a parameter from the full list shown in the "default" file is not found in the user-supplied *zphot.param* file, then the default value is used. All of the parameters that EAZY recognizes and their default values are described below.

3.1.1 Filters

FILTERS_RES	*master.FILTERS.RES*
FILTER_FORMAT	*1*
SMOOTH_FILTERS	*yes*
SMOOTH_SIGMA	*100.*

FILTERS_RES is a text file containing filter response curves, formatted in the same way as the HyperZ filter response file. EAZY is not able to combine multiple entries (e.g. filter×detector) from the response file. Each filter entry consists of a header line showing a) the number of subsequent lines, N_R, that define the filter response and b) a short description of the filter. The header line is followed by N_R lines of three columns: i, λ_i/Å, $R_i(\lambda)$:

```
    ...
  503 41580.0 0.000146
  504 41620.0 0.000116
  505 41650.0 0.000182
427     IRAC_c2 BAND2 total system response (SIRTF/IRAC Web site) 4.5micron (ch2)
  1 37040.0 0.000458
  2 37070.0 0.000398
  3 37090.0 0.000344
    ...
```

FILTER_FORMAT tells the program if the response curves provided in the FILTERS_RES file are determined for energy-counting (FILTER_FORMAT=0) or photon-counting (FILTER_FORMAT=1) detectors. Most modern response curves (via e.g. the ESO ETC webpages or SYNPHOT) are for photon-counting detectors (CCDs), while some older response curves available in the literature may be in the other format (without necessarily saying so; see also Appendix A or the appendix of Maíz Apellániz [2006] for more information).

If SMOOTH_FILTERS is set, then the response curve $R(\lambda)$ is smoothed with a gaussian with width $\sigma =$ SMOOTH_SIGMA$\cdot 1$ Å:

$$\Re_i = \frac{1}{b_i} \sum_{j=1}^{N_R} R_i \cdot \exp\left[\frac{(\lambda_i - \lambda_j)^2}{2\sigma^2}\right], \tag{1}$$

where $b_i = \sum_{j=1}^{N_R} \exp[(\lambda_i - \lambda_j)^2/2\sigma^2]$.

3.1.2 Templates

TEMPLATES_FILE	*../templates/font_nmf.spectra.param*
TEMPLATE_COMBOS	*a*
NMF_TOLERANCE	*1.e-4*

TEMPLATES_FILE is a text file containing a list of the template filenames you wish to use to compute photometric redshifts. The default file is shown below. TEMPLATE_COMBOS specifies the number of templates to combine with non-negative coefficients, α_i, as in Eq. 2 of Brammer, van Dokkum & Coppi [2008].

TEMPLATE_COMBOS =	1	single template fit
	2	pairs of templates as defined in TEMPLATES_FILE
	-2	all pair combinations
	(a or 99)	all templates in TEMPLATES_FILE simultaneously

For fits with single or pairs of templates, α_i are computed analytically following e.g. Bevington & Robinson [2003] (Appendix B). Fitting all of the templates simultaneously, α_i must be computed iteratively, and the iteration tolerance is controlled by the NMF_TOLERANCE parameter. The default value of this parameter reflects a compromise between increasing χ^2 for the template fits when using larger values and increasing computation time for smaller values.

(TEMPLATES_FILE)
```
1   ../templates/font_nmf/font_5_t1.dat    1.0 0 1.0   2,3,4,5
2   ../templates/font_nmf/font_5_t2.dat    1.0 0 1.0   3,4,5
3   ../templates/font_nmf/font_5_t3.dat    1.0 0 1.0   4,5
4   ../templates/font_nmf/font_5_t4.dat    1.0 0 1.0   5
5   ../templates/font_nmf/font_5_t5.dat    1.0 0 1.0
```

The first column is a running count of the files in the template list and is used to identify the templates in the output files. The second column is the relative path from the working directory to the individual template file, which is an ASCII file containing two columns: **1) λ and 2) flux in** F_λ (arbitrarily normalized). The third column is multiplied to the template wavelengths to scale them to Å; for example, if λ in the template file is specified in μm, this column should be $1.e4$. The fourth column is the "age" of the template, in Gyr. If this column is set to something other than zero, then the code will only use this template at redshifts where the age of the universe is greater than the age of the template. The firth column is currently not used by the code. The final column is a comma-separated list of the ID numbers of the other templates that will be combined with the current template if TEMPLATE_COMBOS=2.

WAVELENGTH_FILE	*lambda.def*
TEMP_ERR_FILE	*best_temp_err_v1.out*
TEMP_ERR_A2	*1.00*
SYS_ERR	*0.00*
APPLY_IGM	*y*

All of the templates are rebinned to the (rest-frame) wavelength grid defined in the WAVELENGTH_FILE (Appendix A). The supplied file, *lambda.def*, is a roughly logarithmic grid from 100 Åto 11 μm to allow fits to *rest-frame* NUV and NIR photometry at high- and low-z, respectively. EAZY will ignore redshifts where filters fall off of the template or WAVELENGTH_FILE grids in the observed frame.

TEMP_ERR_FILE is an ASCII file containing the rest-frame template error function described in Brammer, van Dokkum & Coppi [2008]. EAZY reads the first two columns of this file, $\lambda_{\rm te}$ and $\sigma_{\rm te}(\lambda)$. TEMP_ERR_A2 is multiplied to $\sigma_{\rm te}$, so you can easily turn off the effects of the template error function by setting TEMP_ERR_A2=0. SYS_ERR can be used as a minimum fractional error error that is added to the uncertainties equally for every filter and at every redshift. The total flux uncertainty in filter, j, in Eq. 1 of Brammer, van Dokkum & Coppi [2008] is

$$(\delta F_j)^2 = \sigma_{j,{\rm cat}}^2 + F_j^2 \left[\sigma_{\rm sys}^2 + (\text{TEMP\_ERR\_A2} \cdot \sigma_{\rm te}(\lambda_j))^2\right], \tag{2}$$

where $\sigma_{i,{\rm cat}}$ is the flux uncertainty from the photometric catalog and $\sigma_{\rm te}(\lambda_i)$ is the template error interpolated at the rest-frame central wavelength of filter, i.

If APPLY_IGM is set, then IGM absorption in the FUV is applied to the templates following Madau [1995].

DUMP_TEMPLATE_CACHE	*n*
USE_TEMPLATE_CACHE	*n*
CACHE_FILE	*tempfilt.dat*

You can dump the 3D [template][filter][redshift] matrix to a file, CACHE_FILE, to avoid computing it every time you run the code. This is most useful when using lists of hundreds of templates, but be careful when using the template cache files because EAZY is not smart enough to know if the parameters that were used to generate the cache were the same as the parameters of the current run.

3.1.3 Input Files

CATALOG_FILE	*hdfn_fs99_eazy.cat*
NOT_OBS_THRESHOLD	*-90*
N_MIN_COLORS	*5*

CATALOG_FILE is the input photometric catalog that contains photometric fluxes and uncertainties observed in $N_{\rm filt}$ filters. The first line of this file must contain the column names for as many columns as you want the program to read. The column names can be any string, but EAZY will only recognize columns with (case-sensitive) names {*id,z_spec,Fn,En,TOTn*}—see also *zphot.translate*, §3.2. Column, *id*, is just used to identify objects in the output files. Column, *z_spec*, is used when setting FIX_ZSPEC and the values in this column are printed in the *zout* output file. The columns, *Fn* and *En*, are the $\mathbf{F}_\nu$ flux and error columns for filter *n*, respectively. Again, by convention **the photometric fluxes and errors are given in units of F_ν while the template SEDs are provided in F_λ.** The number *n* following *F* or *E* refers to the order of the filters in the FILTERS_RES file. For example, in the HDF-N catalog provided with the EAZY distribution, the second column is the WFPC2 *F300W* flux and the *F300W* filter curve is the 10th entry in the FILTERS_RES file, so the name for this column in the header line should be *F10*. If "color" and "total" fluxes are provided in the photometric catalog (e.g. FIRES), the *TOTn* column can be used to compute the color→total scaling and *n* for the TOT column must match one of the *Fn/En* pairs.

If an object has missing data for some reason in a particular filter, the flux value should be set to some large negative number in the photometric catalog—a number that is more negative than NOT_OBS_THRESHOLD. This value should be more negative than any *measured* negative flux is expected to be, since EAZY naturally handles non-detections with negative fluxes. EAZY will skip any object in the catalog with fewer than N_MIN_COLORS filters with *Fn* > NOT_OBS_THRESHOLD.

3.1.4 Output Files

OUTPUT_DIRECTORY	*OUTPUT*
MAIN_OUTPUT_FILE	*photz*
PRINT_ERRORS	*y*
VERBOSE_LOG	*y*
OBS_SED_FILE	*n*
TEMP_SED_FILE	*n*
POFZ_FILE	*n*
BINARY_OUTPUT	*n*

See §4 for descriptions of the contents of the output files, which are all placed in the OUTPUT_DIRECTORY.

3.1.5 Redshift+Magnitude prior

APPLY_PRIOR	*y*
PRIOR_FILE	*prior_K_zmax7.dat*
PRIOR_FILTER	*28*
PRIOR_ABZP	*25.0*

Set APPLY_PRIOR=*y* to use the redshift/magnitude prior described in Brammer, van Dokkum & Coppi [2008]. The priors are defined in the PRIOR_FILE, which has one column for z and additional columns $P(z|m)$ for each (AB) apparent magnitude bin as defined in the first line of the file. The apparent magnitude in filter $n =$ PRIOR_FILTER is computed for each object from the catalog flux using

$$m_{\mathrm{AB}} = \mathrm{PRIOR\_ABZP} - 2.5 \log_{10} F_n \tag{3}$$

(For catalog fluxes in μJy, PRIOR_ABZP=23.9), and the appropriate $P(z|m_{\mathrm{AB}})$ is chosen and applied as in Eq. 4 of Brammer, van Dokkum & Coppi [2008].

3.1.6 Redshift Grid

FIX_ZSPEC	*n*
Z_MIN	*0.01*
Z_MAX	*4.0*
Z_STEP	*0.01*
Z_STEP_TYPE	*1*

The templates will be fit according to Eqs. 1 and 2 of Brammer, van Dokkum & Coppi [2008] at redshift values in a grid defined by the parameters above. The grid step size will be set following

$$\text{Z\_STEP\_TYPE} = \begin{array}{ll} 0; & \text{step} = \text{Z\_STEP} \\ 1; & \text{step} = \text{Z\_STEP} \times (1+z) \end{array}$$

If FIX_ZSPEC=*y* then EAZY looks for a column, *z_spec*, in the CATALOG_FILE. The template set will then be fit following TEMPLATE_COMBOS to the photometry of each object at the redshift in the redshift grid nearest to *z_spec*.

3.1.7 Zeropoint Offsets

GET_ZP_OFFSETS	*n*
ZP_OFFSET_TOL	*1.e-4*

If GET_ZP_OFFSETS=*y* then EAZY will look for a file, *zphot.zeropoint*, and apply zeropoint offsets to the catalog photometry (§3.3).

3.1.8 Cosmology

H0	*70.0*
OMEGA_M	*0.3*
OMEGA_L	*0.7*

These parameters are used to compute the age of the universe at redshift, z, to compare with the template ages (optionally) specified in the TEMPLATES_FILE. H0 is in units of km/s/Mpc and the OMEGA parameters are the matter and dark energy densities in units of the critical density.

3.2 *zphot.translate*

This file "translates" the column names in CATALOG_FILE to the required F*n* and E*n* formats needed to tell EAZY which filter corresponds to which columns in the catalog (§3.1.3). Using this file allows you to avoid having to edit large catalog files and allows the column names in the catalog to be more meaningful and not tied to a specific FILTERS_RES file. The format of this file is simply two columns, the first being the column name in the CATALOG_FILE and the second containing what you want to "translate" it to (case sensitive). For example, imagine you have a (rather useless) catalog that looks like the following:

```
# id zsp f_f300w e_f300w
1 1.9 1.00 0.100
```

We need to tell EAZY that the 3rd and 4th columns correspond to fluxes and errors measured in the WFPC2 F300W filter (number 10 in our FILTERS_RES file). You could either edit the catalog to look like:

```
# id zsp F10 E10
1 1.9 1.00 0.100
```

or make a *zphot.translate* file like this:

```
 f_f300w F10
e_f300w E10
```

Note that the spectroscopic redshift column isn't required, but if you want EAZY to recognize it it has to be called `z_spec`. So you could add an additional line to the *zphot.translate* file:

`zsp z_spec`.

3.3 *zphot.zeropoint*

While iterative adjustment of photometric zeropoints is not supported in the current version of EAZY you can specify a separate file that will apply zeropoint offsets to the flux and error columns in the the CATALOG_FILE. For example, if you want to multiply the flux and error of filter j by a factor of 1.03 or apply a magnitude offset of +0.07 mag to filter k, you could make a *zphot.zeropoint* file like the following:

```
Fj   1.03
Mk   0.07
```

noting the "F" and "M" as needed for flux or mag offsets and where j, k are the appropriate filter ID numbers from the FILTERS_RES file.

4 Output files

4.1 OUTPUT_FILE.*zout*

This is the main output file where the photometric redshift information is placed. The number of columns in this file depends somewhat on the number of template linear combinations used (TEMPLATE_COMBOS).

- `id`: ID column taken from the input catalog, if it exists
- `z_spec`: Column of spectroscopic redshifts taken from input catalog, if it exists. This is useful for plotting and comparisons to the computed $z_{\rm phot}$, but otherwise not used within EAZY (unless FIX_ZSPEC=Y).
- {`z_1, z_2, z_a`}: Redshift where χ^2 is minimized for the one-, two-, or all-template linear combination modes, *before applying the prior.*
- `z_m1`: Redshift marginalized over $p(z|C) = \exp\{-0.5\chi^2\}$, given by

$$\texttt{z\_m1} = \int z \cdot p(z|C)\ dz, \tag{4}$$

  where $\int p(z|C)dz$ is normalized to unity.
- {`chi_1, chi_2, chi_a`}: (minimum) χ^2 value at $z =$ {`z_1, z_2, z_a`}

- {temp_1, temp_2a/temp_2b}: Best fit template ID numbers at z ={z_1, z_2} for the one- or two- template fits.

- z_p: Redshift where likelihood is maximized (χ^2 is minimized) *after* applying the prior, $p(z|m_0)$.

- chi_p: *Original* χ^2 at z =z_p.

- {temp_p, temp_pa/temp_pb}: Best fit template ID numbers at z =z_p for the one- or two- template fits.

- z_m2: Redshift marginalized over $p(z|C, m_0) = p(z|C) \cdot p(z|m_0)$, given by

$$\texttt{z\_m2} = \int z \cdot p(z|C, m0)\ dz, \tag{5}$$

  which includes the prior and where $\int p(z|C, m_0)\ dz$ is normalized to unity.

  ⋆⋆
  z_m2 usually provides the best photoz estimate and allows you to use a coarser redshift grid, which speeds up the computation significantly. That is, z_m2 computed with Z_STEP=0.03 is usually very similar to z_p computed with Z_STEP=0.005. Note, however that in practice the integral in Eq. 5 assumes that $\lim_{z\to\mathrm{ZMIN}} p(z) = \lim_{z\to\mathrm{ZMAX}} p(z) = 0$. If this isn't true, z_m2 (and z_m1) will be biased from edge effects.
  ⋆⋆

- odds: Redshift quality parameter, $p_{\Delta z}$ or "odds", from Benítez [2000], that represents the fraction of the total integrated probability that lies within $\pm\Delta z$ of the zphot estimate, and is designed to identify sources that have broad and/or multi-modal probability distributions. Here we use $\Delta z = 0.2$.

- {l68, u68, l95, u95, l99, u99}: If PRINT_ERRORS=y then these columns are printed, representing the 1,2, and 3σ confidence intervals computed from the $p(z)$ probability distributions [§2.5, Brammer, van Dokkum & Coppi, 2008].

- nfilt: Number of filters, j, used in the fit, with flux,

$$\mathrm{F}j > \mathrm{NOT\_OBS\_THRESHOLD} \tag{6}$$

4.2 OUTPUT_FILE.*param*

This file is created if VERBOSE_LOG=1 and it contains all of the parameter values from *zphot.param* used to compute OUTPUT_FILE.*zout*, as well as information on the individual filters and templates used. This file itself can be used to recompute the photozs just by copying it to *zphot.param* in your working directory.

4.3 *[ID].obs_sed*

If OBS_SED_FILE=y then an ascii file will be created for each object in the catalog containing the catalog and best-fit template fluxes in each filter, with columns

- `lambda`: central wavelength of the (normalized) filter transmission,

$$\texttt{lambda} = \int \lambda \cdot R(\lambda)\ d\lambda \tag{7}$$

- `flux_cat`: object flux taken directly from the catalog
- `err_full`: total error used in the fit (Eq.2)
- `temp[1,2,a]_z`: normalized template fluxes integrated through the filters at z =`z_[1,2,a]` (best redshift *without* the prior)
- `temp[1,2,a]_zprior`: normalized template fluxes integrated through the filters at z =`z_p` (best redshift *with* the prior)

4.4 *[ID].temp_sed*

If TEMP_SED_FILE=*y* then an ascii file will be created for each object in the catalog containing the full redshifted template SEDs converted to F_ν, normalized to the object fluxes, and including the IGM absorption if APPLY_IGM=*y*. You should be able to plot these SEDs directly on top of the (F_ν) catalog fluxes. The columns in this file are

- `lambda`: wavelength as defined in WAVELENGTH_FILE multiplied by $(1+$ `z_[1,2,a]`$)$
- `tempflux`: full best-fit SED at $z =$ `z_[1,2,a]`.
- `lambda_zprior`: wavelength as defined in WAVELENGTH_FILE multiplied by $(1+$ `z_p`$)$
- `tempflux_zprior`: full best-fit SED at $z =$ `z_p`

NB: The best-fit normalization coefficients for each template in TEMPLATES_FILE are given in the header of the *temp_sed* file.

4.5 *[ID].pz*

If POFZ_FILE=*y* then an ascii file will be created for each object in the catalog containing $p(z)$. The columns of this file are:

- `z`: redshift as computed from the grid parameters
- `chi2`: χ^2 from the fit at redshift, z.
- `prior`: $p(z|m_0)$ extracted from PRIOR_FILE, printed if APPLY_PRIOR=*y*.
- `pz`: $p(z|C, m_0)$, printed if APPLY_PRIOR=*y*.
- {`t1, t2a/t2b`}: Best-fit template ID numbers at redshift, z, printed if TEMPLATE_COMBOS=*1* or *2*.

4.6 BINARY_OUTPUT=y

For even modest nubers of catalog objects, writing the *obs_sed*, *temp_sed*, and *pz* files for each object can quickly take up large amounts of disk space and the file IO can significantly slow down the execution of the code, to say nothing about having thousands of files to deal with. All of these problems can be addressed by writing this output information into binary files from which all of the same information can be extracted as from the ASCII files. Setting BINARY_OUTPUT=y will create a number of binary files in the output directory in lieu of the ascii files described above. The binary files can be read by IDL for plotting, etc. An IDL program file is created in the output directory, [].*readbin.pro*, that demonstrates how to read the binary files. See the auxiliary file, *sed.pro*, for an example on how to overplot best-fit template SEDs on top of observed fluxes.

NB: For a test catalog of 1000 objects and 6 filters, running EAZY with all of the ASCII outputs set took 27 s and created 96 Mb in 3000 output files. In contrast, with BINARY_OUTPUT=y, EAZY took 19 s to run and created 1.8 Mb in six output files.

References

Benítez, N. 2000, ApJ, 536, 571

Bevington, P. R., & Robinson, D. K. 2003, Data reduction and error analysis for the physical sciences, 3rd ed., by Philip R. Bevington, and Keith D. Robinson. Boston, MA: McGraw-Hill, ISBN 0-07-247227-8, 2003.,

Blaizot, J., Wadadekar, Y., Guiderdoni, B., Colombi, S. T., Bertin, E., Bouchet, F. R., Devriendt, J. E. G., & Hatton, S. 2005, MNRAS, 360, 159

Bolzonella, M., Miralles, J.-M., & Pelló, R. 2000, A&A, 363, 476

Brammer, G., van Dokkum, P. G., & Coppi, P. 2008, ApJ, submitted

De Lucia, G., & Blaizot, J. 2007, MNRAS, 375, 2

Fernández-Soto, A., Lanzetta, K. M., & Yahil, A. 1999, ApJ, 513, 34

Fioc, M., & Rocca-Volmerange, B. 1997, A&A, 326, 950

Madau, P. 1995, ApJ, 441, 18

Maíz Apellániz, J. 2006, AJ, 131, 1184

Sha, F., Lin, Y., Saul, L. K., & Lee, D. D. 2007, Neural Computation, 19, 2004

A Interpolation and integration

All of the integrations of the filter fluxes, probability distributions, etc. are done using a simple implementation of the trapezoid rule. We interpolate the user-defined template SEDs (with arbitrary wavelength sampling) to a common (rest-frame) wavelength grid defined in the WAVELENGTH_FILE, using a robust interpolation algorithm that preserves flux, shown in Figure 1. In the

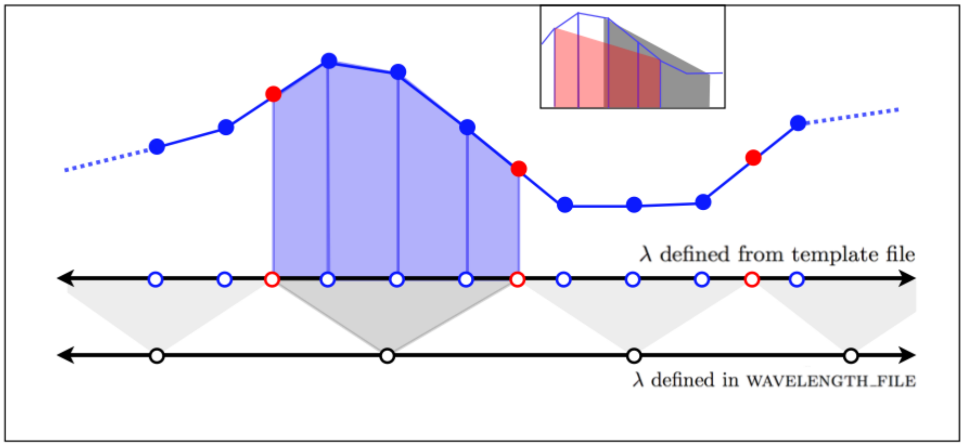

Figure 1: Interpolating the input template spectra to the master wavelength grid defined in the WAVELENGTH_FILE.

figure, the black circles are the semi-regular wavelength points defined in the file specified by the WAVELENGTH_FILE parameter, and the red points are the midpoints of this wavelength grid. The blue points represent the (arbitrary) wavelength grid of a user-supplied template file, which, in the case of synthetic templates, is generally more finely sampled than the WAVELENGTH_FILE grid. We use the trapezoid rule (shaded blue regions) to integrate between the linear-interpolated midpoints and the adjacent template wavelength point, between the intervening template points, and finally between the last template wavelength point and the next midpoint. If there are no template points between the midpoints, we simply integrate between the midpoints. The inset shows two other simpler integration schemes that only integrate between the master grid points. If the curvature is significant between the master wavelength points, the interpolated flux can significantly under- or overestimate the true template fluxes. Systematic errors in the integrated template fluxes of order a few percent can result in systematic errors in the derived photometric redshifts.

B Linear combinations

EAZY provides the option of fitting (positive) linear combinations of the templates specified in the TEMPLATES_FILE (§3.1.2) to the object photometry defined in the catalog. The normalization coefficients for either one or two template fits are calculated analytically, while the normalizations for "N"-template fits are fit iteratively following the algorithm of Sha et al. [2007].

B.1 Single template fit

The normalization of each template, α_i, at redshift, z, is computed from the sum over NFILT filters, j, following

$$\alpha_i = \frac{\sum_j \left(T_{z,i,j} \cdot F_j\right) / \left(\delta F_j\right)^2}{\sum_j T_{z,i,j}^2 / \left(\delta F_j\right)^2}, \tag{8}$$

where $T_{z,i,j}$ is the integrated template flux and δF_j is the full error as defined in Eq. 2.

B.2 Two template fit

The two-template normalizations for templates i_1 and i_2 are computed analytically using least squares as follows.

$$\mathcal{T} = \begin{bmatrix} T_{z,i_1,0} & T_{z,i_1,1} & \dots & T_{z,i_1,\mathrm{NFILT}} \\ T_{z,i_2,0} & T_{z,i_2,1} & \dots & T_{z,i_2,\mathrm{NFILT}} \end{bmatrix}, \quad \mathcal{F} = \begin{bmatrix} F_0 & F_1 & \dots & F_{\mathrm{NFILT}} \end{bmatrix} \tag{9}$$

$$A = \mathcal{T}\mathcal{T}^{\mathrm{T}}, \quad \mathbf{b} = \mathcal{T} \cdot \mathcal{F} \tag{10}$$

and

$$(\alpha_{i_1},\ \alpha_{i_2}) = A^{-1}\mathbf{b}\ . \tag{11}$$

Note that each of the template (T) and observed (F) fluxes above in filter j is weighted by $1/\delta F_j^2$. If one of the template amplitudes, α_{i_1} or α_{i_2}, is negative, then the best *single* template fit is used at redshift, z.

B.3 N template fit

We use the remarkably simple algorithm of Sha et al. [2007] to compute the *non-negative* template normalizations for NTEMP > 2 template fits.

$$\mathcal{T} = \begin{bmatrix} T_{z,0,0} & T_{z,0,1} & \dots & T_{z,0,\mathrm{NFILT}} \\ T_{z,1,0} & T_{z,1,1} & \dots & T_{z,1,\mathrm{NFILT}} \\ \vdots & \vdots & \ddots & \vdots \\ T_{z,\mathrm{NTEMP},0} & T_{z,\mathrm{NTEMP},1} & \dots & T_{z,\mathrm{NTEMP},\mathrm{NFILT}} \end{bmatrix}, \quad \mathcal{F} = \begin{bmatrix} F_0 & F_1 & \dots & F_{\mathrm{NFILT}} \end{bmatrix}. \tag{12}$$

Again each template and observed flux is weighted by the uncertainty in filter j.

$$A = \mathcal{T}\mathcal{T}^{\mathrm{T}}, \quad \mathbf{b} = -\ \mathcal{T} \cdot \mathcal{F} \tag{13}$$

We start by setting all of the normalization coefficients to unity,

$$\mathbf{c} = (\alpha_1, \alpha_2, \dots) = (1, 1, \dots)\,. \tag{14}$$

And we compute multiplicative updates to the coefficients by

$$\mathbf{d} = A\ \mathbf{c}, \tag{15}$$

$$c_{i,\mathrm{new}} = c_{i,\mathrm{old}} \cdot \frac{b_i}{d_i}, \tag{16}$$

iterating while

$$\frac{\sum_i |c_{i,\mathrm{new}} - c_{i,\mathrm{old}}|}{\sum_i c_{i,\mathrm{old}}} > \text{NMF\_TOLERANCE}\ . \tag{17}$$

The best-fit χ^2 value generally decreases for smaller values of NMF_TOLERANCE, but the best-fit *redshifts* don't generally change much for NMF_TOLERANCE $< 10^{-3}$.