Single burst age and metallicity from Hβ, Mg b and $\langle$Fe$\rangle$ for model and observed elliptical galaxies

P. Polko

Kapteyn Institute, Department of Astronomy, Groningen University, Groningen

polko@astro.rug.nl

S.C. Trager

Kapteyn Institute, Department of Astronomy, Groningen University, Groningen

sctrager@astro.rug.nl

and

R.S. Somerville

Archive Branch, Space Telescope Science Institute, Baltimore

somerville@stsci.edu

October 14, 2005

Abstract

In this paper a method is presented to calculate the Single Starburst Population (SSP) age and metallicity directly from the spectrum of an observed galaxy, using the Lick/IDS index system. A stellar population model is used to supply the calibration. The method is applied to elliptical and S0 galaxies in a Coma-like halo from the semi-analytic Somerville model and to twelve observed galaxies in the actual Coma cluster. The results are compared to see whether the model galaxies have the same properties as the observed galaxies. The SSP age and metallicity values of the model galaxies are also compared with ages and metallicities as calculated by the model itself.

The SSP ages correlate best with the epoch of the last major merger, although there is significant scatter and the SSP age is systematically about two gigayears higher, due to the light of the older stars. The SSP metallicities best reflect the stellar-light-weighted metallicity. The model metallicities are much lower than the derived metallicities of the Coma galaxies, while the model ages are higher, except for the $\langle$Fe$\rangle$ index. At present the method does not give very precise results between the three indices, although with another stellar population model these three SSP values could converge.

1 Introduction

One of the long-standing goals in astronomy is to derive the stellar populations inside a galaxy. This would give information on many properties of galaxies, and thereby constrain the current models of galaxy formation. A step in that direction is the determination of the Single Starburst Population (SSP) values, where the galaxy is approximated by a population of stars which all have the same age and metallicity. Despite the simplification, these values offer a convenient way to characterize galaxies and compare results with other studies. A problem in deriving these values is the age-metallicity degeneracy. A change of a factor 3 in age of a stellar population has the same effects as a change of a factor 2 in its metallicity on the metal-line strength and optical colours (Worthey 1994). This paper continues in the spirit of Worthey (1994) and Trager et al. (2000a) to use a spectral index more sensitive to age and one more sensitive to metallicity to lift the degeneracy. When an age-sensitive index is plotted versus a metallicity-sensitive index calculated for model stars with known age and metallicity, they establish a two dimensional grid.

When the same indices of an observed galaxy are plotted on top of the grid, the SSP age and metallicity of that galaxy can be deduced. When speaking of the age and metallicity of a galaxy, these SSP values are meant.

To see how well the method works, we use the Somerville semi-analytic model of galaxy formation (Somerville & Primack 1999, hereafter SP99; Somerville, Primack & Faber 2001; Somerville et al. 2004). This model generates star formation histories with specific metallicities of galaxies in a dark matter halo, and the corresponding mean stellar-mass-weighted age and metallicity and the time since the last major merger. In addition to these properties we compute the mean stellar-light-weighted age and metallicity from the star formation history and mass-to-light ratios provided by the model. From the star formation histories we calculate the present day spectra of the galaxy, using the Bruzual & Charlot (2003) stellar population model. We use our method to calculate the ages and metallicities from the spectra and compare them with the aforementioned properties.

This article has the following structure. In section 2 the origins of the data used in this project are described. In section 3 the methods used are explained. In section 4 the results are presented. Section 5 covers the conclusions and comparison with previous work. The method and results are discussed in section 6.

2 Data

The data used in this project come from three sources. The stellar population model grid is from the Bruzual & Charlot model (Bruzual & Charlot 2003, henceforth BC03), the model galaxies are the result of the semi-analytic Somerville model (SP99) and the observed galaxies are from Trager et al. (2005). In this section these sources are discussed in more detail.

2.1 Bruzual & Charlot model

The BC03 model uses the isochrone synthesis technique to compute the spectral evolution of stellar populations. It uses a stellar evolution model to find the isochrone, consisting of stars with different masses, corresponding to a certain age and metallicity and a stellar spectral library to assign spectra along the isochrone. An initial mass function (IMF) is assumed and used to integrate over mass. The specific model used here comprises the Padova 1994 stellar evolution prescription and the Chabrier IMF (Chabrier 2001). For further information on the model see BC03.

Assuming a star formation history and a metal-enrichment law, it is possible to calculate how many stars of which metallicity have formed up until a certain time. By adding the integrated spectra with corresponding age and metallicity together, the spectrum of the stellar population at that time can be computed.

In this paper only the spectra integrated over the isochrones are used, equivalent to a delta function in the star formation history and metal enrichment. The grid used to determine age and metallicity consists of 14 chosen ages (1, 1.5, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 Gyr) and the six available metallicities (0.005, 0.02, 0.2, 0.4, 1, 2.5 $Z_{\odot}$). Spectra of these ages and metallicities were extracted from the model (e.g. figure 1).

2.2 Semi-analytic models

Theoretical simulation of galaxy formation and evolution can offer great insight into the fundamental processes that take place. Two different approaches have come into existence.

One is the N-body simulation. A very large number of particles is arranged according to an initial configuration and allowed to be influenced by other particles and possibly an invariable background. Physical laws can be added for a more realistic result.
However a fundamental problem of this approach is the limitation in scale imposed by the limits of computing power. It is, and will be for some time, impossible to simulate particles at the (sub)stellar and cluster scale simultaneously, including all the accompanying physics in order to form galaxies. (Peacock 2001).

The second approach is the semi-analytic model. The extended Press-Schechter formalism (Bond et al. 1991; Bower 1991; Lacey & Cole 1993) is used to obtain a history of merging halos. All the mass in a halo is divided in states (e.g. hot gas, cold gas, stars) with certain properties (e.g. metallicity, time since last merger). According to (approximations

of) physical formulae the gas is constantly redistributed over the different states. In the case of a merger, the mass in the two or more merging halos is added and again redistributed following a set of rules. In this way a lot of (observable) quantities, like the star formation history and disc sizes, can be extracted from the simulation. Since this model stores only the total mass in a state and calculates just fractions of this mass, it is much faster than an N-body simulation.
This method suffers from a few problems as well. Every aspect needs expressions that are usually experimentally not well known or cannot even be observed. They also supply the researcher with a vast array of parameters that need to be set. Some of these are taken from N-body simulations, some are set to simulate observational constraints like the Tully-Fisher relation, others represent a personal preference. Even so they still have trouble fitting all known data. For example, in the models, due to continuing gas infall and late mergers of gas-rich galaxies, larger galaxies are bluer than small ones, contrary to observations (Kauffmann, White & Guiderdoni 1993; Cole et al. 1994; Somerville, Primack & Faber 2001). One way to alleviate this last problem would be to have more efficient star formation in large galaxies. After some time, the star formation could be halted by preventing the gas to cool, for example through a black hole that has formed in the centre. The large galaxy would passively evolve and become redder, while the small galaxy would continue to produce stars at a slow rate, remaining blue.

2.3 Somerville model

The Somerville model, although developed independently, continues in the spirit of the models by Kauffman, White & Guiderdoni (1993) and Cole et al. (1994), often called the Munich and Durham groups respectively, because of their affiliation with those universities. The data used in this project is the output of several runs of the model.

One run consists of 10 realisations of a Coma sized halo with a circular velocity of 1200 km s^{-1}. The cosmological parameters are $\Omega_m = 0.3$, $\Omega_\lambda = 0.7$, and $H_0 = 70$ km s^{-1} Mpc^{-1}. Of the 4234 galaxies in the 10 realisations, 713 have Hubble type E or S0, as determined from the bulge-to-disk ratio (see SP99 for more information on how bulges and disks are assigned). Galaxies with bulge-to-disk ratios below 0.68 were classified as spiral, between 0.68 and 1.52 as an S0, and over 1.52 as elliptical. Galaxies without bulges were categorised as irregulars. For more information on the model see SP99.

2.4 Observed galaxies

The observational data has been obtained from Trager et al. (2005), who observed twelve early-type galaxies in the Coma cluster (Abell 1656) with the Low Resolution Imaging Spectrograph (LRIS) (Oke et al. 1995) on the Keck Telescopes. See the paper for more detailed information.

3 Method

3.1 Lick/IDS indices

The Lick/IDS system was developed by Burstein et al. (1984) and Faber et al. (1985) and originally consisted of 11 indices. The system was expanded by Worthey et al. (1994) to include a total of 21 indices and the wavelength definitions were later refined by Trager et al. (1998). The indices used in this paper (table 1) measuring atomic absorption lines, have remained the same since the Worthey et al. paper.
In order to calculate an index, a continuum around the wavelength of interest is needed. The local continuum, denoted by $F_C(\lambda)$, is approximated by a straight line passing through two points at each side of the index. These two points are calculated as the average continuum in a small wavelength range:

$$F_p = \frac{1}{\lambda_2 - \lambda_1} \int_{\lambda_1}^{\lambda_2} F(\lambda) \mathrm{d}\lambda \qquad (1)$$

with λ_1 and λ_2 of the left and right side of the index given by the column Pseudocontinua in table 1. The equivalent width used is then calculated as:

$$\mathrm{EW} = \int_{\lambda_1}^{\lambda_2} \left(1 - \frac{F(\lambda)}{F_C(\lambda)}\right) \mathrm{d}\lambda \qquad (2)$$

with wavelength, flux and equivalent width in Ångström and λ_1 and λ_2 given by the column Index Bandpass. The central bandpass is shown in figure 1.

Table 1: Index definitions.

Number	Name	Index Bandpass	Pseudocontinua	Units
9	Hβ	4847.875-4876.625	4827.875-4847.875	Å
			4876.625-4891.625	
13	Mgb	5160.125-5192.625	5142.625-5161.375	Å
			5191.375-5206.375	
14	Fe5270	5245.650-5285.650	5233.150-5248.150	Å
			5285.650-5318.150	
15	Fe5335	5312.125-5352.125	5304.625-5315.875	Å
			5353.375-5363.375	

Fractional pixels have been dealt with by performing an integration over a quadratic fit based on the adjacent bins.

Since the Somerville models are based on the BC03 models, they have the same resolution and a systemic velocity of zero.

To derive the parameters of the twelve observed galaxies, the BC03 model spectra need the same resolution and redshift as the galaxy spectra, so the systemic velocity and the velocity dispersion of the galaxies are needed.
In order to obtain these values, each BC03 spectrum has been fitted to each galaxy by minimising $\chi^2 = |G - P_C(B(v,\sigma) \circ M)|^2$. Here G is the galaxy spectrum, P_C is a polynomial, $B(v,\sigma)$ is a broadening function (depending on the systemic velocity v and the velocity dispersion σ), which is convolved with a stellar template M. The polynomial was not used in the rest of the calculation, because it changes the ratios between the index strengths. The BC03 spectrum with the smallest χ^2 then gives the systemic velocity and velocity dispersion for each galaxy (Trager et al. 2005).
For each galaxy the BC03 model spectra have been smoothed with the velocity dispersion. After that, the model spectra have been degraded to a full width-half maximum value of 4.4 Å, which is the resolution of the LRIS spectra of Coma galaxies. The wavelengths of the resulting spectra have been multiplied by $\frac{(1+v/c)}{(1-v/c)}$ before the indices were calculated. These indices establish the grid on which the indices derived directly from the galaxy spectrum can be compared.

3.2 Deriving parameters

Mg b, $\langle$Fe$\rangle$ (=Fe5270/2+Fe5335/2) and [MgFe] ($=\sqrt{\text{Mg } b\langle\text{Fe}\rangle}$) were used as metallicity indicators. The combined index $\langle$Fe$\rangle$ was used because its error is smaller than the error in its constituents and [MgFe] was used because it is stable to variations in [Mg/Fe] (Trager et al. 2000a, Thomas et al. 2003). These indices, along with Hβ, an age indicator, were calculated for each BC03 spectrum and when Hβ is plotted against another index they establish a grid. The same indices were calculated for (model) galaxies and an automated program transformed a quadrangular cell with a galaxy to a square, checked whether galaxies were inside the square and calculated the interpolated age and metallicity of those galaxies. This procedure cannot be used if the galaxy lies in a region where the grid folds back on itself, which occurs in some grids (as for example in figure 2). As this grid folding is a result of the BC03 models only, better sampling of the grid or another stellar population model could solve this problem.

4 Results

4.1 General observations

The first striking feature in the BC03 models are the trends in age and metallicity, stressing the age-metallicity degeneracy, as can be seen in figure 1. The flux goes down in this wavelength region as the age and metallicity go up. Furthermore the Mg b index strength increases with increasing age and metallicity, but more with increasing metallicity. The same is true for Fe5270 and Fe5335. Hβ

decreases with increasing age and metallicity, but is more sensitive to age.

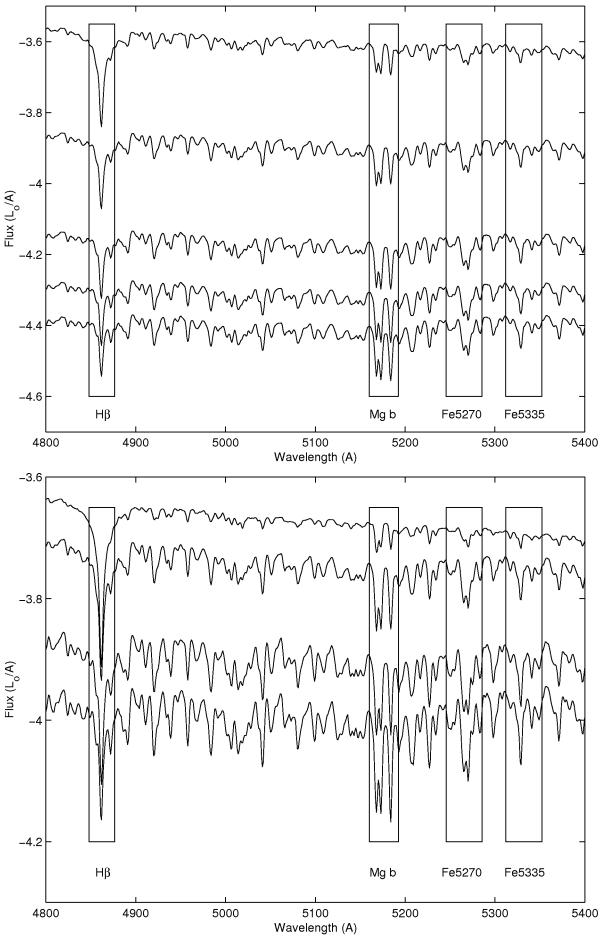

Figure 1: Age-metallicity degeneracy. In the upper plot, the spectra of ages 1, 2, 4, 6 and 8 Gyr (top to bottom) are given for a metallicity of 0.2 $Z_{\odot}$. In the lower plot are the metallicities 0.005, 0.2, 1 and 2.5 $Z_{\odot}$ (top to bottom) for an age of 1.5 Gyr. Overplotted are the central bandpasses of the relevant indices.

4.2 Somerville galaxies

An example of the BC03 grid with galaxy overplotted can be seen in figure 2, which shows that most of the galaxies lie in the non-overlapping part of the grid. The galaxies that are outside the grid have not been included in the results. Galaxies that are in regions where multiple ages or metallicities could be assigned, have arbitrarily been assigned the highest possible metallicity and corresponding age, or in case of equal metallicities, the highest age. In figures 3 and 4 the spectral age is plotted

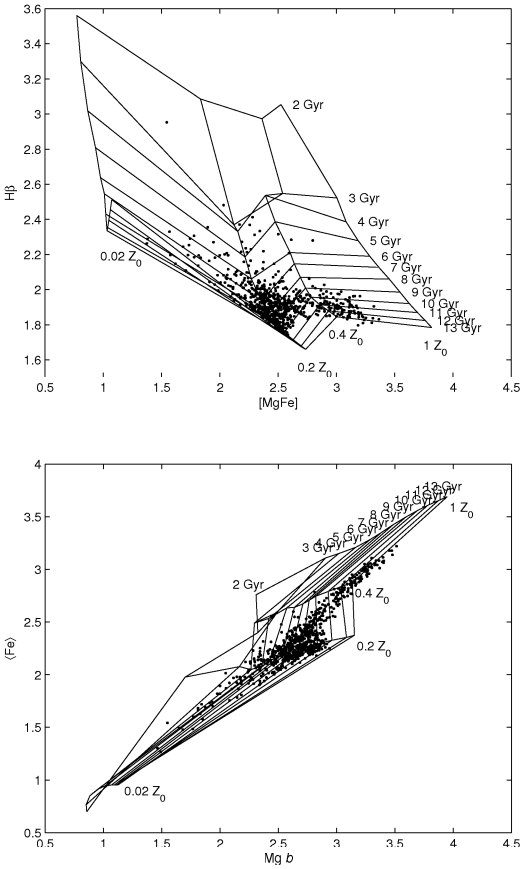

Figure 2: BC03 grids with Coma-like galaxies from Somerville superimposed. Only the relevant part is shown and labeled for clarity. Ages are in gigayears and metallicities in $Z_{\odot}$.

versus the mean stellar-mass-weighted age and the time since the last major merger the galaxy was involved in as provided by the Somerville model. The computed mean stellar-light-weighted age is also included. This last quantity has been calculated for each galaxy from the models by using

$$\frac{\sum_{i,j} t_i M_{ij} \left[(\frac{M}{L})_{ij}\right]^{-1}}{\sum_{i,j} M_{ij} \left[(\frac{M}{L})_{ij}\right]^{-1}}. \qquad (3)$$

Here i sums over age, j sums over metallicity, t_i is the time since formation of the galaxy, M_{ij} is the mass at time t_i and metallicity j that is converted into stars and $(\frac{M}{L})_{ij}$ is the corresponding mass to light ratio.

The tracer with the slope closest to that of the spectral age of a galaxy is the time since the last major merger, although the amount of scatter is high and the spectral age is about two gigayears higher. This corroborates Trager et al. (2000a), who claim that the SSP age of a galaxy is sensitive to recent star formation which is associated with mergers. That the spectral age is higher than the merger indicates that previous star formation (i.e., the older populations) influence the SSP age.

The mean stellar-light-weighted metallicity has a slope that is almost equal to that of the spectral metallicity in log space, although the spectral metallicity is about 0.25 dex smaller. The clumping around the spectral metallicity of -0.7 is probably due to the discontinuity in the grid at 0.2 $Z_\odot$.

In figure 5 the histograms of age and metallicity are given. As can be seen the age distribution is broad with ages between 3 and 13 Gyr, while the metallicity distribution covers only a small part of possible values (10^{-5} to 2 $Z_\odot$), with metallicities between 0.1 and 0.7 $Z_\odot$. This is in agreement with the findings of Trager et al. (2000a).

4.3 Observed Coma galaxies

In table 2 the results for the 12 Coma galaxies are given. In the second column are the three indices against which Hβ is plotted. In many Mg b–Hβ plots the galaxy fell beyond the high-metallicity side of the grid. In some cases it was close enough to warrant extrapolation. If so, the cell which would include the galaxy, when linearly extrapolated to higher metallicity, was used to obtain the values. NGC 4874 fell below the highest age (13 Gyr) in the $\langle$Fe$\rangle$ index, so the approximate metallicity is very uncertain. In calculations this galaxy has been given the exact age of 13 Gyr and metallicity of 0.7 $Z_\odot$ for the $\langle$Fe$\rangle$ index.

The surprisingly good determination of GMP 3565 might be due to the low velocity dispersion of the galaxy. Other low σ galaxies (D154, D158) have a low spread as well. High-σ galaxies also have very strong Mg b lines, characteristic of high metallicity stars and non-solar [α/Fe].

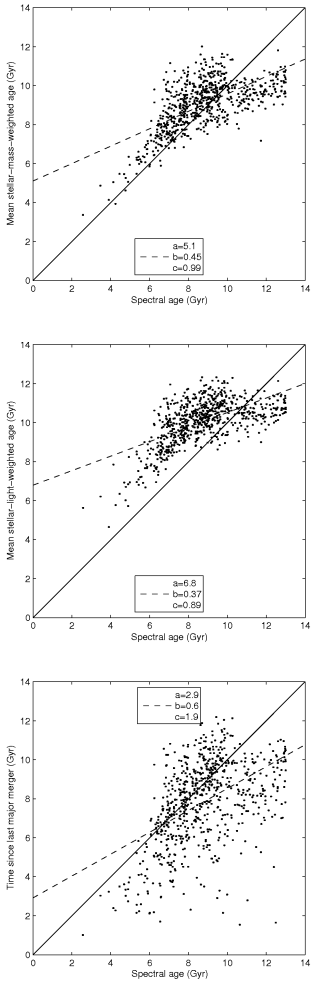

Figure 3: Possible tracers of age. The full line is for reference and each dot is one galaxy. In the upper panel the mean stellar-mass-weighted age is plotted. In the middle panel is the time since the last major merger. In the lower panel the computed mean stellar-light-weighted age is shown. The offset a, the slope b and the standard deviation σ of the relation are given in all plots. Spectral ages plotted here are determined from [MgFe].

Table 2: Parameters for Coma galaxies.

galaxy	index	t (Gyr)	Z ($Z_\odot$)	extra. (%)	σ_t	σ_Z	v (km/s)	σ (km/s)	χ^2
D127	Mg b	X	X		3.85	1.22	7467	110	2.3
	$\langle$Fe$\rangle$	8.7	0.94						
	[MgFe]	3.3	2.67	(12)[a]					
D128	Mg b	3.0	2.98	(32)[a]	3.02	1.12	7967	104	6.5
	$\langle$Fe$\rangle$	9.0	0.77						
	[MgFe]	5.7	1.60						
D154	Mg b	3.3	2.09		1.58	0.56	6726	51	1.4
	$\langle$Fe$\rangle$	6.3	0.97						
	[MgFe]	4.0	1.64						
D157	Mg b	4.0	2.58	(05)[a]	4.44	1.03	6086	111	5.1
	$\langle$Fe$\rangle$	12.1	0.66						
	[MgFe]	11.3	0.96						
D158	Mg b	4.1	1.31		0.82	0.38	5976	60	2.0
	$\langle$Fe$\rangle$	5.6	0.56						
	[MgFe]	5.4	0.82						
GMP 3565	Mg b	5.1	0.85		0.04	0.02	7142	46	1.1
	$\langle$Fe$\rangle$	5.2	0.81						
	[MgFe]	5.1	0.83						
NGC 4864	Mg b	X	X		6.46	1.37	6735	211	3.9
	$\langle$Fe$\rangle$	13.3	0.75						
	[MgFe]	4.2	2.69	(13)[a]					
NGC 4867	Mg b	X	X		X	X	4784	223	7.8
	$\langle$Fe$\rangle$	9.2	0.92						
	[MgFe]	X	X						
NGC 4871	Mg b	X	X		5.49	1.31	6684	162	10.0
	$\langle$Fe$\rangle$	11.7	0.84						
	[MgFe]	3.9	2.69	(12)[a]					
NGC 4872	Mg b	X	X		5.35	1.37	7169	211	38.0
	$\langle$Fe$\rangle$	11.3	0.79						
	[MgFe]	3.8	2.74	(16)[a]					
NGC 4873	Mg b	X	X		4.55	1.24	5750	169	4.3
	$\langle$Fe$\rangle$	10.3	0.75						
	[MgFe]	3.8	2.51	(01)[a]					
NGC 4874	Mg b	X	X		7.52	1.34	7142	274	17.7
	$\langle$Fe$\rangle$	>13	$\pm$0.7						
	[MgFe]	5.4	2.60	(06)[a]					

X means a good determination was impossible.
[a]Extrapolated value using adjacent cell,
(percentage of linear extrapolation).

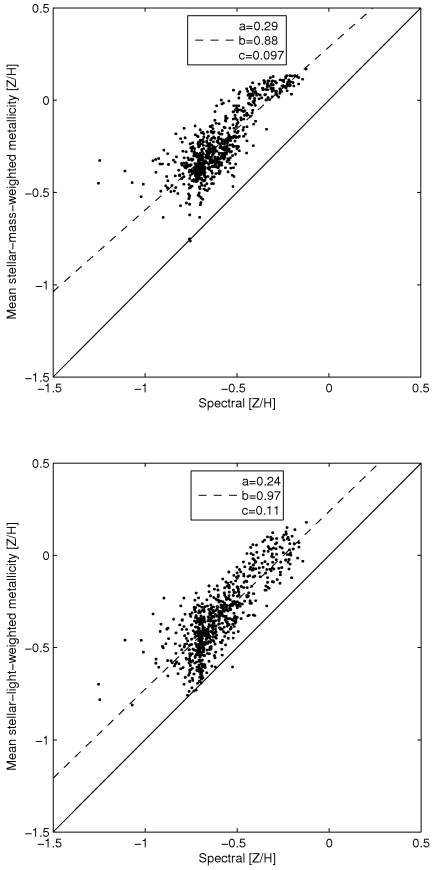

Figure 4: Possible tracers of metallicity. The full line is for reference and each dot is one galaxy. In the upper panel the mean stellar-mass-weighted metallicity is plotted. In the lower panel the computed mean stellar-light-weighted metallicity is shown. The offset a, the slope b and the standard deviation σ of the relation are given in all plots. Spectral metallicities plotted here are determined from [MgFe].

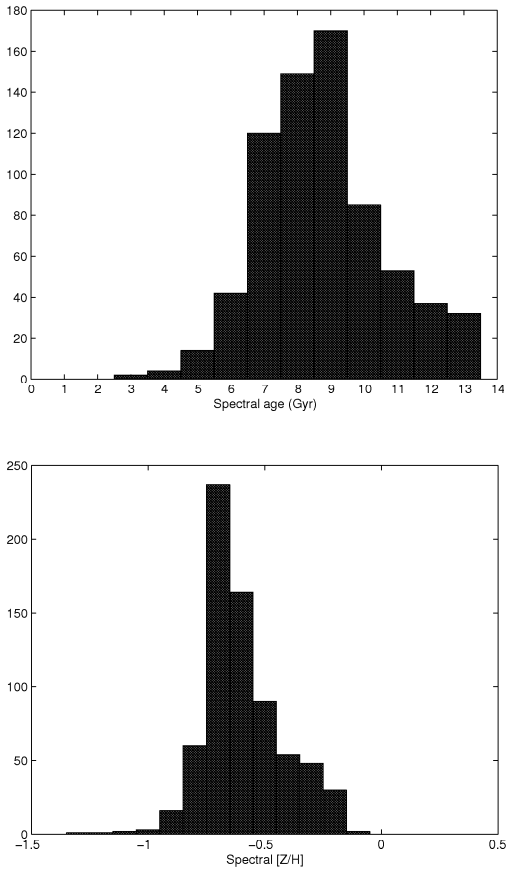

Figure 5: Age and metallicity histograms of the 713 model galaxies in a Coma-like halo as determined from [MgFe].

Another feature is the fact that if the spectral age is high, the corresponding metallicity is low and vice versa, representing the high metallicity end of the grid, where increasing age and metallicity are almost parallel.

4.4 Comparison

As is clear from figure 6 the Mg b index and derivative [MgFe] index give ages for the model galaxies that are too high and a metallicity value that is too low in comparison with the observed elliptical galaxies. The $\langle$Fe$\rangle$ index has the correct spread in age, but, again, the metallicity from the models is too low compared to the observed galaxies.

Figure 7 shows that the spread in the difference between Mg b and $\langle$Fe$\rangle$ metallicities of the model can only account for about 30% of the same difference in the observed galaxies, so elliptical galaxies do have [Mg/Fe]$>$ 0. By projecting the ages and metallicities derived from Mg b and $\langle$Fe$\rangle$ on the lower panel of figure 2, Mg b lying above $\langle$Fe$\rangle$ values, it can be seen that the low age-high metallicity Fe values are the results of the Fe indices being depressed in comparison with the Mg index, arriving at the same conclusion. As indicated in table A1 of the appendix of BC03 the predictions for line strengths around 2.5 $Z_{\odot}$ are only fair, so the difference caused by the Bruzual & Charlot models could be diminished by better [Mg/Fe] model prescriptions.

5 Conclusions & Discussion

The spectral age correlates best with the time since the last major merger of a galaxy, despite the scatter and the 2 Gyr higher spectral age. Both the mean stellar-mass-weighted and mean stellar-light-weighted metallicities are well approximated by the spectral metallicity, despite the systematic underestimation by 0.25 dex. The difference in stellar-mass-weighted and stellar-light-weighted metallicity is much smaller than the difference in age. This is due to the fact that the luminosity of a star is highly dependant on its mass through temperature. The metallicity, acting through the opacity, has a comparatively small role.

The fact that the Fe indices are depressed in comparison with the Mg indices has has been reported

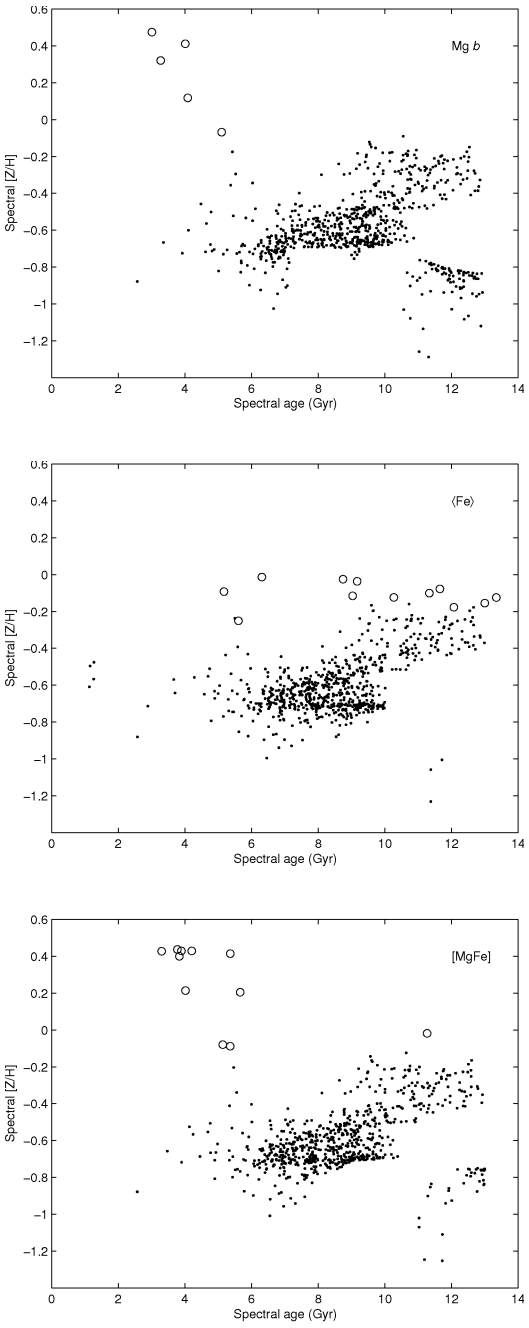

Figure 6: Comparison of Somerville galaxies with Coma galaxies. The plots are labeled with the index used to determine age and metallicity. In all plots open circles are Coma galaxies and dots are model galaxies. The four galaxies younger than 2 Gyr and probably all of the galaxies in the lower right corner of all indices, where the grids always overlap (figure 2), are the wrong choice of multiple-valued determinations of age and metallicity.

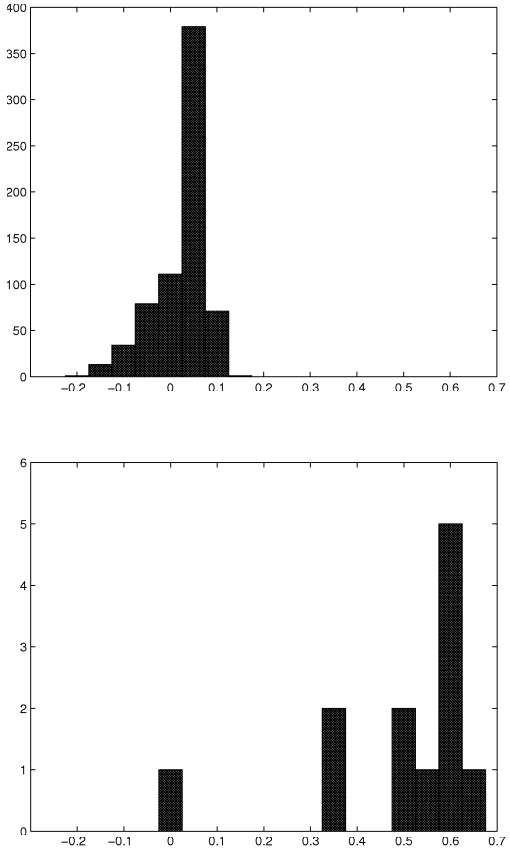

Figure 7: Histograms of the difference between metallicities as determined from the Mg b and $\langle$Fe$\rangle$ indices for the model galaxies (upper panel) and the observed galaxies (lower panel). Where an Mg b determination of the metallicity was impossible, a value of 3 $Z_{\odot}$ was assumed.

before by Vazdekis et al.(1996), Greggio (1997) and Trager et al. (2000a), who also show mathematically why Fe is depressed rather than Mg enhanced. Worthey et al. (1992) give possible physical causes for this problem. The reason for the poor agreement between the Mg b and $\langle$Fe$\rangle$ values for the observed Coma galaxies lie in the BC03 models, since they do not take [Mg/Fe] ratios other than the solar value into account. This would also explain why the method seems to work best if the galaxy has a small velocity dispersion. Trager et al. (2000b) show [Mg/Fe] is tightly correlated with the velocity dispersion, meaning low-σ galaxies have metallicities close to solar. These would be fit better by the BC03 models and produce similar ages and metallicities from the indices.

The method as presented here does however suffer from a few problems. The grid as taken from the BC03 models sometimes overlaps itself, causing ambiguities in some age-metallicity regions. The metallicity grid is rather coarse with only six values spanning three orders of magnitude. A finer grid would make the estimation of age and metallicity more precise. It might also cause the aforementioned ambiguities to vanish, as a straight line connecting two grid points and crossing other lines might in reality curve around them (see the top of the enlarged part in figure 2 for example). One straightforward improvement would be to check if the galaxy is in the gridcell before the transformation. This would cause a few galaxies to get a better estimate of age and metallicity.

A more fundamental problem lies in the fact that stellar populations very nearly add up as light-weighted vectors in the grid (Trager et al. 2000b), while the grid itself is curved. Suppose a galaxy consists of two populations, one 13 Gyr old at 0.2 $Z_{\odot}$, the other also 13 Gyr but at 1 $Z_{\odot}$. The top panel in figure 2 shows that the combined populations would be assigned an age from the [MgFe] index that is much higher than 13 Gyr. The same process can be applied to the metallicities as well. A population of 3 Gyrs at 0.4 $Z_{\odot}$ combined with a population of 5 Gyrs at 0.4 Gyrs would be assigned a metallicity above 0.4 $Z_{\odot}$. Since the grid lines of constant metallicity are straighter than those of constant age in the region of interest, this problem affects age more than metallicity. For populations spanning many ages and metallicities an almost rectangular grid is needed to produce values that

faithfully represent the different populations. Certain combinations of indices, like [MgFe] for metallicity, could improve the rectangularity of the grid. Alternatively, another stellar population model can be used. The Vazdekis model (Vazdekis 1999) produces a grid which has less jolts and is nearly rectangular over a larger region in age-metallicity space than the BC03 grid.

6 References

Bond, J. R., Cole, S., Efstathiou, G., & Kaiser, N. 1991, ApJ, 301, 27
Bower., R. 1991, MNRAS, 248, 332
Bruzual, G., & Charlot, S. 2003, MNRAS, 344, 1000
Burstein, D., Faber, S.M., Gaskell, C.M., & Krumm, N. 1984, Apj, 287, 586
Chabrier, G. 2001, Apj, 554, 1274
Cole, S., Aragón-Salamanca, A., Frenk, C.S., Navarro, J.F., & Zepf, S.E. 1994, MNRAS, 271, 781
Faber, S.M., Friel, E.D., Burstein, D., & Gaskell, C.M. 1985, ApJS, 57, 711
Greggio, L. 1997, MNRAS, 285, 151
Kauffmann, G., White, S.D.M., & Guiderdoni, B. 1993, MNRAS, 264, 201
Lacey, C., & Cole, S. 1993, MNRAS, 262, 627
Peacock, J.A. 2001, Cosmological Physics, third reprint, Cambridge University Press
Somerville, R.S., & Primack, J.R. 1999, MNRAS, 310, 1087
Somerville, R.S., Primack, J.R., & Faber, S.M. 2001, MNRAS, 320, 504
Somerville, R.S., Moustakas, L.A., Mobasher, B., Gardner, J.P., Cimatti, A., Conselice, C., Daddi, E., Dahlen, T., Dickinson, M., Eisenhardt, P., Lotz, J., Papovich, C., Renzini, A., & Stern, D. 2004, Apj, 600, L135
Thomas, Maraston, Bender, 2003, MNRAS, 399, 897
Trager, S.C., Worthey, G., Faber, S.M., Burstein, D., & González, J.J. 1998, ApJS, 116, 1
Trager, S.C., Faber, S.M., Worthey, G., & González, J.J. 2000a, AJ, 119, 1645
Trager, S.C., Faber, S.M., Worthey, G., & González, J.J. 2000b, AJ, 120, 165
Trager, S.C., Worthey, G., Faber, S.M., & Dressler, A. 2005, MNRAS, 362, 2
Vazdekis, A., Casuso, E., Peletier, R. F., & Beckman, J. E. 1996, ApJS, 106, 307
Vazdekis, A. 1999, ApJ, 513, 224
Worthey, G., Faber, S.M., & González, J.J. 1992, ApJ, 398, 69
Worthey, G. 1994, ApJS, 95, 107

A Program code

```
//Made by Peter Polko in February 2004
//Last change: Tuesday 21 September 2004, added age and metallicity bounds
//Compile with: gcc -o programname programname.c -lm (on Unix-systems)

//Libraries
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<math.h>

//Fixed values
#define NUMBERSMAX  2000    //Maximum number of galaxies to be processed
#define AGES        17      //Maximum number of ages to be read
#define METS        6       //Maximum number of metallicities to be read
#define INDS        20      //Maximum number of indices to be read

//Global variables
float   ages[17]={1.015e+09,1.434e+09,2.0e+09,3.0e+09,4.0e+09,5.0e+09,6.0e+09,
                  7.0e+09,8.0e+09,9.0e+09,1.0e+10,1.1e+10,1.2e+10,1.3e+10,1.4e+10,
                  1.5e+10,1.6e+10}; //Ages used
float   metallicities[6]={0.0001,0.0004,0.004,0.008,0.02,
                          0.05}; //Available metallicities
char    inputfilename[64],outputfilename[64];
FILE    *inputfile,*outputfile; //File for output
char    *text[28];

//Prototypes
int     main(void);
void    interpolation(void);
void    trans(double a[],double x[],double y[],double p[],double q[]);
void    snart(double u[],double v[],double a[],double x1[],double y1[],int max,
              int plus);
FILE    *openfile(char *filename,char *mode);
void    closefile(FILE *file,char *filename);

//Functions
int     main(void)
{
    text[0]="Give file with calibration indices: ";
    text[1]="Give file with indices: ";
    text[2]="Choose index (Mgb=16,[MgFe]=28,<Fe>=29): ";
    text[3]="Choose lowest age bound (Gyr)  [5]: ";
    text[4]="Choose highest age bound (Gyr) [13]: ";
    text[5]="Choose lowest metallicity bound (Zo)  [2]: ";
    text[6]="Choose highest metallicity bound (Zo) [5]: ";
    text[7]="Give filename for output: ";
    text[8]="Wrong index. Quitting.\n";
    text[9]="Can't open file \"%s\".\n";
    text[10]="File \"%s\" opened.\n";
    text[11]="File \"%s\" closed.\n";

    interpolation();
```

```
    return(0);
}

//The interpolation program.
void    interpolation(void)
{
    int     i,j,k,galaxymax;
    int     index1,index2=12/*Hbeta*/;
    int     indmgb=16,indfe1=17,indfe2=18,indfe=28,indmgfe=29;
//Index numbers of Mg b, Fe5270, Fe5335, <Fe> and [MgFe]
    int     minage,maxage,minmet,maxmet;
    float   BC[AGES][METS][INDS],indices[NUMBERSMAX][INDS];
    double  BC1[AGES][METS],BC2[AGES][METS],indices1[NUMBERSMAX],indices2[NUMBERSMAX];
    double  indicesx[NUMBERSMAX],indicesy[NUMBERSMAX];
    double  a[8],x[4],y[4],p[4]={0,1,1,0},q[4]={0,0,1,1};
    double  x1[1],y1[1];
    double  themet[NUMBERSMAX],theage[NUMBERSMAX];
    char    kar;
    char    names[30][7]={"HdA","HdF","CN1","CN2","Ca4227","G4300","HgA","HgF",
                          "Fe4383","Ca4455","Fe4531","Fe4668","Hb","Fe5015","Mg1",
                          "Mg2","Mgb","Fe5270","Fe5335","Fe5406","Fe5709","Fe5782",
                          "NaD","TiO1","TiO2","Ca1","Ca2","Ca3","[MgFe]","<Fe>"};


//Open file with B&C model indices
    printf(text[0]);scanf("%s",inputfilename);
    inputfile=openfile(inputfilename,"r");

//Read model indices
    kar=getc(inputfile);
    while(kar=='#' || kar=='%')
    {
        while((kar=getc(inputfile))!='\n'){}
        kar=getc(inputfile);
    }
    ungetc(kar,inputfile);
    for(i=0;i<METS;i++)
        for(j=0;j<AGES;j++)
            for(k=0;k<INDS;k++)
                fscanf(inputfile,"%f",&BC[j][i][k]);
    closefile(inputfile,inputfilename);

//Open file with indices
    printf(text[1]);scanf("%s",inputfilename);
    inputfile=openfile(inputfilename,"r");

//Read indices
    kar=getc(inputfile);
    while(kar=='#' || kar=='%')
    {
        while((kar=getc(inputfile))!='\n'){}
        kar=getc(inputfile);
    }
```

```
    ungetc(kar,inputfile);
    for(i=0;fscanf(inputfile,"%f",&indices[i][0])==1;i++)
        for(j=1;j<INDS;j++)
            fscanf(inputfile,"%f",&indices[i][j]);
    galaxymax=i;
    closefile(inputfile,inputfilename);

//Choose boundaries of age and metallicity
    for(i=0;i<AGES;i++)
        printf("%d\t%g\n",i,ages[i]/1e9);
    printf(text[3]);scanf("%d",&minage);
    printf(text[4]);scanf("%d",&maxage);
    for(i=0;i<METS;i++)
        printf("%d\t%g\n",i,metallicities[i]/0.02);
    printf(text[5]);scanf("%d",&minmet);
    printf(text[6]);scanf("%d",&maxmet);

//Open output file
    printf(text[7]);scanf("%s",outputfilename);
    outputfile=openfile(outputfilename,"w");

//Choose index
    for(i=0;i<INDS;i++)
        printf("%d\t%s\n",i,names[i]);
    printf("%d\t%s\n",28,names[28]);
    printf("%d\t%s\n",29,names[29]);
    printf(text[2]);scanf("%d",&index1);

    if(index1<INDS)
    {
        for(i=0;i<AGES;i++)
            for(j=0;j<METS;j++)
            {
                BC1[i][j]=BC[i][j][index1];
                BC2[i][j]=BC[i][j][index2];
            }
        for(i=0;i<galaxymax;i++)
        {
            indices1[i]=indices[i][index1];
            indices2[i]=indices[i][index2];
        }
    }
    else if(index1==indfe)
    {
        for(i=0;i<AGES;i++)
            for(j=0;j<METS;j++)
            {
                BC1[i][j]=(BC[i][j][indfe1]+BC[i][j][indfe2])/2;
                BC2[i][j]=BC[i][j][index2];
            }
        for(i=0;i<galaxymax;i++)
        {
            indices1[i]=(indices[i][indfe1]+indices[i][indfe2])/2;
            indices2[i]=indices[i][index2];
```

```
        }
    }
    else if(index1==indmgfe)
    {
        for(i=0;i<AGES;i++)
            for(j=0;j<METS;j++)
            {
                BC1[i][j]=sqrt(BC[i][j][indmgb]*(BC[i][j][indfe1]+
                                                 BC[i][j][indfe2])/2);
                BC2[i][j]=BC[i][j][index2];
            }
        for(i=0;i<galaxymax;i++)
        {
            indices1[i]=sqrt(indices[i][indmgb]*(indices[i][indfe1]+
                                                 indices[i][indfe2])/2);
            indices2[i]=indices[i][index2];
        }
    }
    else
    {
        printf(text[8]);
        exit(0);
    }

    for(i=minage;i<maxage;i++)   //Choose closest range in age and
        for(j=minmet;j<maxmet;j++)//metallicity to prevent ambiguities
        {
            for(k=0;k<galaxymax;k++)
                indicesx[k]=indicesy[k]=-1;
            x[0]=BC1[i+1][j];x[1]=BC1[i+1][j+1];x[2]=BC1[i][j+1];x[3]=BC1[i][j];
            y[0]=BC2[i+1][j];y[1]=BC2[i+1][j+1];y[2]=BC2[i][j+1];y[3]=BC2[i][j];
//Transform (0,0) forth and back to see which solution is correct
            trans(a,p,q,x,y);
            snart(x,y,a,x1,y1,1,1);
//Choose the correct solution and use it to obtain age and metallicity
            if(x1[0]>-0.1 && x1[0]<0.1)
                snart(indices1,indices2,a,indicesx,indicesy,galaxymax,1);
            else
                snart(indices1,indices2,a,indicesx,indicesy,galaxymax,0);
            for(k=0;k<galaxymax;k++)
                if(indicesx[k]>=0 && indicesx[k]<1 && indicesy[k]>=0 && indicesy[k]<1)
                {
                    themet[k]=metallicities[j]+indicesx[k]*(metallicities[j+1]-
                                                            metallicities[j]);
                    theage[k]=(ages[i]+(1-indicesy[k])*(ages[i+1]-ages[i]));
                }
        }
    for(i=0;i<galaxymax;i++)
    {
      fprintf(outputfile,"%.5f\t%.5f\n",theage[i]/1e9,themet[i]/0.02);
      //Ages outputted in Gyr, metallicities in solar metallicity
    }
    closefile(outputfile,outputfilename);
}
```

```
//Transformation from an arbitrary quadrangle to an arbitrary quadrangle.
void    trans(double a[],double x[],double y[],double p[],double q[])
{
    a[0]=((p[3]*x[1]*y[0]*y[2]-p[1]*y[0]*x[3]*y[2]-x[1]*p[3]*y[1]*y[2]+x[3]*p[1]*y[3]*
     y[2]+y[3]*y[0]*x[2]*p[1]-p[1]*y[3]*x[2]*y[2]+x[1]*y[1]*y[3]*p[2]-y[3]*x[1]*y[0]*
     p[2]-y[1]*y[0]*x[2]*p[3]+p[3]*y[1]*x[2]*y[2]-x[3]*y[3]*y[1]*p[2]+y[1]*y[0]*x[3]*
     p[2])*x[0]-x[1]*y[1]*x[2]*y[3]*p[0]-p[3]*x[1]*y[0]*x[2]*y[2]+x[1]*p[0]*x[3]*y[1]*
     y[2]+x[3]*y[3]*x[2]*y[1]*p[0]+x[3]*y[0]*x[1]*y[3]*p[2]-x[3]*p[0]*y[1]*x[2]*y[2]-
     x[3]*p[0]*x[1]*y[3]*y[2]-x[1]*y[0]*x[3]*y[1]*p[2]-x[3]*y[3]*y[0]*x[2]*p[1]+x[1]*
     p[0]*y[3]*x[2]*y[2]+p[1]*y[0]*x[3]*x[2]*y[2]+x[1]*y[1]*y[0]*x[2]*p[3])/((-y[3]*
     x[3]*y[1]+x[1]*y[3]*y[1]-x[1]*y[3]*y[0]+x[2]*y[3]*y[0]-y[3]*x[2]*y[2]+y[1]*x[2]*
     y[2]-y[2]*x[3]*y[0]-x[1]*y[2]*y[1]+x[3]*y[1]*y[0]+y[2]*x[3]*y[3]+x[1]*y[2]*y[0]-
     x[2]*y[1]*y[0])*x[0]-x[1]*y[3]*x[2]*y[1]+x[1]*y[2]*x[3]*y[1]+x[1]*y[3]*x[2]*y[2]-
     x[1]*y[2]*x[3]*y[3]+x[1]*y[0]*x[3]*y[3]+x[2]*y[1]*x[3]*y[3]-x[1]*y[0]*x[3]*y[1]+
     x[1]*y[0]*x[2]*y[1]-x[1]*y[0]*x[2]*y[2]+y[0]*x[3]*x[2]*y[2]-y[0]*x[2]*x[3]*y[3]-
     x[3]*y[1]*x[2]*y[2]);
    a[1]=-((y[0]*y[3]*p[1]+p[3]*y[0]*y[2]-p[1]*y[0]*y[2]+y[1]*y[0]*p[2]-y[1]*y[0]*
     p[3]-y[3]*y[0]*p[2])*x[0]-x[3]*y[3]*y[1]*p[2]-x[3]*y[0]*y[3]*p[1]-x[1]*y[0]*y[1]*
     p[2]+p[3]*y[1]*x[2]*y[2]-p[1]*y[3]*x[2]*y[2]+x[1]*y[0]*y[1]*p[3]+x[1]*y[1]*y[3]*
     p[2]+x[3]*y[0]*y[3]*p[2]-x[1]*p[3]*y[1]*y[2]+x[1]*p[0]*y[1]*y[2]-x[3]*p[0]*y[3]*
     y[2]+x[3]*p[1]*y[3]*y[2]+p[1]*y[0]*x[2]*y[2]-p[0]*y[1]*x[2]*y[2]+p[0]*y[3]*x[2]*
     y[2]-p[3]*y[0]*x[2]*y[2]+x[3]*p[0]*y[3]*y[1]-x[1]*p[0]*y[3]*y[1])/((-y[3]*x[3]*
     y[1]+x[1]*y[3]*y[1]-x[1]*y[3]*y[0]+x[2]*y[3]*y[0]-y[3]*x[2]*y[2]+y[1]*x[2]*y[2]-
     y[2]*x[3]*y[0]-x[1]*y[2]*y[1]+x[3]*y[1]*y[0]+y[2]*x[3]*y[3]+x[1]*y[2]*y[0]-x[2]*
     y[1]*y[0])*x[0]-x[1]*y[3]*x[2]*y[1]+x[1]*y[2]*x[3]*y[1]+x[1]*y[3]*x[2]*y[2]-x[1]*
     y[2]*x[3]*y[3]+x[1]*y[0]*x[3]*y[3]+x[2]*y[1]*x[3]*y[3]-x[1]*y[0]*x[3]*y[1]+x[1]*
     y[0]*x[2]*y[1]-x[1]*y[0]*x[2]*y[2]+y[0]*x[3]*x[2]*y[2]-y[0]*x[2]*x[3]*y[3]-x[3]*
     y[1]*x[2]*y[2]);
    a[2]=((-y[1]*x[1]*p[2]-x[3]*p[1]*y[3]-x[1]*y[0]*p[3]+y[3]*x[3]*p[2]+x[1]*p[3]*
     y[1]-x[3]*y[0]*p[2]+p[3]*y[0]*x[2]+p[1]*x[2]*y[2]-p[3]*x[2]*y[2]+x[1]*y[0]*p[2]+
     x[3]*y[0]*p[1]-p[1]*y[0]*x[2])*x[0]+x[3]*p[0]*x[1]*y[3]+x[3]*p[0]*x[2]*y[2]+x[3]*
     y[3]*x[2]*p[1]-x[1]*p[0]*x[3]*y[1]-x[1]*p[0]*x[2]*y[2]-x[1]*y[1]*x[2]*p[3]-x[3]*
     y[3]*x[2]*p[0]+x[1]*x[3]*y[1]*p[2]+p[3]*x[1]*x[2]*y[2]+x[1]*y[1]*x[2]*p[0]-x[3]*
     x[1]*y[3]*p[2]-p[1]*x[3]*x[2]*y[2])/((-y[3]*x[3]*y[1]+x[1]*y[3]*y[1]-x[1]*y[3]*
     y[0]+x[2]*y[3]*y[0]-y[3]*x[2]*y[2]+y[1]*x[2]*y[2]-y[2]*x[3]*y[0]-x[1]*y[2]*y[1]+
     x[3]*y[1]*y[0]+y[2]*x[3]*y[3]+x[1]*y[2]*y[0]-x[2]*y[1]*y[0])*x[0]-x[1]*y[3]*x[2]*
     y[1]+x[1]*y[2]*x[3]*y[1]+x[1]*y[3]*x[2]*y[2]-x[1]*y[2]*x[3]*y[3]+x[1]*y[0]*x[3]*
     y[3]+x[2]*y[1]*x[3]*y[3]-x[1]*y[0]*x[3]*y[1]+x[1]*y[0]*x[2]*y[1]-x[1]*y[0]*x[2]*
     y[2]+y[0]*x[3]*x[2]*y[2]-y[0]*x[2]*x[3]*y[3]-x[3]*y[1]*x[2]*y[2]);
    a[3]=((-p[1]*y[2]-y[3]*p[2]+p[3]*y[2]-y[1]*p[3]+y[3]*p[1]+y[1]*p[2])*x[0]+y[3]*
     x[1]*p[2]+p[0]*y[3]*x[2]-x[1]*y[0]*p[2]+x[1]*y[0]*p[3]-x[1]*p[0]*y[3]+x[3]*p[0]*
     y[1]+p[1]*x[3]*y[2]-p[3]*x[1]*y[2]+x[1]*p[0]*y[2]-y[3]*x[2]*p[1]-p[3]*y[0]*x[2]-
     p[0]*x[3]*y[2]+y[1]*x[2]*p[3]-x[3]*y[0]*p[1]+x[3]*y[0]*p[2]-y[1]*x[3]*p[2]-p[0]*
     y[1]*x[2]+p[1]*y[0]*x[2])/((-y[3]*x[3]*y[1]+x[1]*y[3]*y[1]-x[1]*y[3]*y[0]+x[2]*
     y[3]*y[0]-y[3]*x[2]*y[2]+y[1]*x[2]*y[2]-y[2]*x[3]*y[0]-x[1]*y[2]*y[1]+x[3]*y[1]*
     y[0]+y[2]*x[3]*y[3]+x[1]*y[2]*y[0]-x[2]*y[1]*y[0])*x[0]-x[1]*y[3]*x[2]*y[1]+x[1]*
     y[2]*x[3]*y[1]+x[1]*y[3]*x[2]*y[2]-x[1]*y[2]*x[3]*y[3]+x[1]*y[0]*x[3]*y[3]+x[2]*
     y[1]*x[3]*y[3]-x[1]*y[0]*x[3]*y[1]+x[1]*y[0]*x[2]*y[1]-x[1]*y[0]*x[2]*y[2]+y[0]*
     x[3]*x[2]*y[2]-y[0]*x[2]*x[3]*y[3]-x[3]*y[1]*x[2]*y[2]);
    a[4]=((-q[1]*y[0]*x[3]*y[2]-x[1]*q[3]*y[1]*y[2]+q[3]*x[1]*y[0]*y[2]+x[3]*q[1]*
     y[3]*y[2]+y[3]*y[0]*x[2]*q[1]-q[1]*y[3]*x[2]*y[2]+x[1]*y[1]*y[3]*q[2]-y[3]*x[1]*
     y[0]*q[2]+y[1]*y[0]*x[3]*q[2]+q[3]*y[1]*x[2]*y[2]-x[3]*y[3]*y[1]*q[2]-y[1]*y[0]*
     x[2]*q[3])*x[0]-x[3]*q[0]*x[1]*y[3]*y[2]+q[1]*y[0]*x[3]*x[2]*y[2]+x[3]*y[3]*x[2]*
```

```
     y[1]*q[0]+x[3]*y[0]*x[1]*y[3]*q[2]-x[1]*y[0]*x[3]*y[1]*q[2]-x[1]*y[1]*x[2]*y[3]*
     q[0]+x[1]*q[0]*x[3]*y[1]*y[2]-x[3]*q[0]*y[1]*x[2]*y[2]+x[1]*q[0]*y[3]*x[2]*y[2]-
     q[3]*x[1]*y[0]*x[2]*y[2]+x[1]*y[1]*y[0]*x[2]*q[3]-x[3]*y[3]*y[0]*x[2]*q[1])/
     ((-y[3]*x[3]*y[1]+x[1]*y[3]*y[1]-x[1]*y[3]*y[0]+x[2]*y[3]*y[0]-y[3]*x[2]*y[2]+
     y[1]*x[2]*y[2]-y[2]*x[3]*y[0]-x[1]*y[2]*y[1]+x[3]*y[1]*y[0]+y[2]*x[3]*y[3]+x[1]*
     y[2]*y[0]-x[2]*y[1]*y[0])*x[0]-x[1]*y[3]*x[2]*y[1]+x[1]*y[2]*x[3]*y[1]+x[1]*y[3]*
     x[2]*y[2]-x[1]*y[2]*x[3]*y[3]+x[1]*y[0]*x[3]*y[3]+x[2]*y[1]*x[3]*y[3]-x[1]*y[0]*
     x[3]*y[1]+x[1]*y[0]*x[2]*y[1]-x[1]*y[0]*x[2]*y[2]+y[0]*x[3]*x[2]*y[2]-y[0]*x[2]*
     x[3]*y[3]-x[3]*y[1]*x[2]*y[2]);
    a[5]=((-y[0]*y[3]*q[1]+q[1]*y[0]*y[2]-q[3]*y[0]*y[2]-y[1]*y[0]*q[2]+y[1]*y[0]*
     q[3]+y[3]*y[0]*q[2])*x[0]+x[3]*y[3]*y[1]*q[2]+x[3]*y[0]*y[3]*q[1]+x[1]*y[0]*y[1]*
     q[2]-x[1]*y[0]*y[1]*q[3]+q[1]*y[3]*x[2]*y[2]-q[3]*y[1]*x[2]*y[2]-x[1]*y[1]*y[3]*
     q[2]-x[3]*y[0]*y[3]*q[2]+x[3]*q[0]*y[3]*y[2]+x[1]*q[3]*y[1]*y[2]-x[3]*q[1]*y[3]*
     y[2]-x[1]*q[0]*y[1]*y[2]+q[3]*y[0]*x[2]*y[2]+q[0]*y[1]*x[2]*y[2]-q[0]*y[3]*x[2]*
     y[2]-q[1]*y[0]*x[2]*y[2]-q[0]*y[3]*x[3]*y[1]+x[1]*q[0]*y[3]*y[1])/((-y[3]*x[3]*
     y[1]+x[1]*y[3]*y[1]-x[1]*y[3]*y[0]+x[2]*y[3]*y[0]-y[3]*x[2]*y[2]+y[1]*x[2]*y[2]-
     y[2]*x[3]*y[0]-x[1]*y[2]*y[1]+x[3]*y[1]*y[0]+y[2]*x[3]*y[3]+x[1]*y[2]*y[0]-x[2]*
     y[1]*y[0])*x[0]-x[1]*y[3]*x[2]*y[1]+x[1]*y[2]*x[3]*y[1]+x[1]*y[3]*x[2]*y[2]-x[1]*
     y[2]*x[3]*y[3]+x[1]*y[0]*x[3]*y[3]+x[2]*y[1]*x[3]*y[3]-x[1]*y[0]*x[3]*y[1]+x[1]*
     y[0]*x[2]*y[1]-x[1]*y[0]*x[2]*y[2]+y[0]*x[3]*x[2]*y[2]-y[0]*x[2]*x[3]*y[3]-x[3]*
     y[1]*x[2]*y[2]);
    a[6]=-((x[3]*q[1]*y[3]-x[1]*q[3]*y[1]-y[3]*x[3]*q[2]-q[1]*x[2]*y[2]+x[1]*y[0]*
     q[3]-x[1]*y[0]*q[2]+q[3]*x[2]*y[2]+y[1]*x[1]*q[2]-x[3]*y[0]*q[1]+q[1]*y[0]*x[2]+
     x[3]*y[0]*q[2]-q[3]*y[0]*x[2])*x[0]+x[1]*q[0]*x[3]*y[1]+x[3]*x[1]*y[3]*q[2]-x[3]*
     q[0]*x[2]*y[2]-x[3]*q[0]*x[1]*y[3]-x[1]*x[3]*y[1]*q[2]+q[1]*x[3]*x[2]*y[2]-x[1]*
     y[1]*x[2]*q[0]+x[3]*y[3]*x[2]*q[0]-x[3]*y[3]*x[2]*q[1]-q[3]*x[1]*x[2]*y[2]+x[1]*
     q[0]*x[2]*y[2]+x[1]*y[1]*x[2]*q[3])/((-y[3]*x[3]*y[1]+x[1]*y[3]*y[1]-x[1]*y[3]*
     y[0]+x[2]*y[3]*y[0]-y[3]*x[2]*y[2]+y[1]*x[2]*y[2]-y[2]*x[3]*y[0]-x[1]*y[2]*y[1]+
     x[3]*y[1]*y[0]+y[2]*x[3]*y[3]+x[1]*y[2]*y[0]-x[2]*y[1]*y[0])*x[0]-x[1]*y[3]*x[2]*
     y[1]+x[1]*y[2]*x[3]*y[1]+x[1]*y[3]*x[2]*y[2]-x[1]*y[2]*x[3]*y[3]+x[1]*y[0]*x[3]*
     y[3]+x[2]*y[1]*x[3]*y[3]-x[1]*y[0]*x[3]*y[1]+x[1]*y[0]*x[2]*y[1]-x[1]*y[0]*x[2]*
     y[2]+y[0]*x[3]*x[2]*y[2]-y[0]*x[2]*x[3]*y[3]-x[3]*y[1]*x[2]*y[2]);
    a[7]=-((y[1]*q[3]-y[1]*q[2]-q[3]*y[2]-y[3]*q[1]+y[3]*q[2]+q[1]*y[2])*x[0]+q[0]*
     y[1]*x[2]-y[3]*x[1]*q[2]+x[1]*q[0]*y[3]-x[1]*y[0]*q[3]+x[1]*y[0]*q[2]-x[1]*q[0]*
     y[2]+q[0]*x[3]*y[2]-x[3]*y[0]*q[2]+q[3]*y[0]*x[2]+x[3]*y[0]*q[1]+y[3]*x[2]*q[1]-
     q[1]*y[0]*x[2]-y[1]*x[2]*q[3]-q[1]*x[3]*y[2]-q[0]*x[3]*y[1]-q[0]*y[3]*x[2]+y[1]*
     x[3]*q[2]+q[3]*x[1]*y[2])/((-y[3]*x[3]*y[1]+x[1]*y[3]*y[1]-x[1]*y[3]*y[0]+x[2]*
     y[3]*y[0]-y[3]*x[2]*y[2]+y[1]*x[2]*y[2]-y[2]*x[3]*y[0]-x[1]*y[2]*y[1]+x[3]*y[1]*
     y[0]+y[2]*x[3]*y[3]+x[1]*y[2]*y[0]-x[2]*y[1]*y[0])*x[0]-x[1]*y[3]*x[2]*y[1]+x[1]*
     y[2]*x[3]*y[1]+x[1]*y[3]*x[2]*y[2]-x[1]*y[2]*x[3]*y[3]+x[1]*y[0]*x[3]*y[3]+x[2]*
     y[1]*x[3]*y[3]-x[1]*y[0]*x[3]*y[1]+x[1]*y[0]*x[2]*y[1]-x[1]*y[0]*x[2]*y[2]+y[0]*
     x[3]*x[2]*y[2]-y[0]*x[2]*x[3]*y[3]-x[3]*y[1]*x[2]*y[2]);
}

//The reverse transformation which unfortunately has two solutions.
void    snart(double u[],double v[],double a[],double x1[],double y1[],int max,
              int plus)
{
    int     i;
    double  A,B,C,D;

    if((a[1]*a[7]-a[3]*a[5])!=0)
    {
        A=a[3]*(a[3]*a[6]-a[2]*a[7])/(a[1]*a[7]-a[3]*a[5]);
```

```
        for(i=0;i<max;i++)
        {
            B=(a[3]*(a[3]*(a[4]-v[i])+a[7]*(u[i]-a[0])+a[1]*a[6]-a[2]*a[5]))/(a[1]*
              a[7]-a[3]*a[5]);
            C=(a[3]*(a[1]*(a[4]-v[i])+a[5]*(u[i]-a[0])))/(a[1]*a[7]-a[3]*a[5]);

            if(A==0)
            {
                if(B==0)
                {
                    if(C==0)
                        printf("Problem with transformatieon: A=B=C=0.\n");
                    else
                        printf("Problem with transformation: A=B=0,C!=0.\n");
                }
                else
                {
                    y1[i]=-C/B;
                    x1[i]=(u[i]-a[0]-a[2]*y1[i])/(a[1]+a[3]*y1[i]);
                }
            }
            else
            {
                D=B*B-4*A*C;
                if(D<0)
//                  printf("Problem with transformation: D<0.\n");
                else if(D==0)
                {
                    y1[i]=-B/(2*A);
                    x1[i]=(u[i]-a[0]-a[2]*y1[i])/(a[1]+a[3]*y1[i]);
                }
                else
                {
                    if(plus)
                    {
                        y1[i]=(-B+sqrt(D))/(2*A);
                        x1[i]=(u[i]-a[0]-a[2]*y1[i])/(a[1]+a[3]*y1[i]);
                    }
                    else
                    {
                        y1[i]=(-B-sqrt(D))/(2*A);
                        x1[i]=(u[i]-a[0]-a[2]*y1[i])/(a[1]+a[3]*y1[i]);
                    }
                }
            }
        }
        else if((a[2]*a[7]-a[3]*a[6])!=0)
        {
            for(i=0;i<max;i++)
            {
                y1[i]=(a[3]*(a[4]-v[i])+a[7]*(u[i]-a[0]))/(a[2]*a[7]-a[3]*a[6]);
                x1[i]=(u[i]-a[0]-a[2]*y1[i])/(a[1]+a[3]*y1[i]);
            }
```

```
    }
    else
        printf("Help, problem with transformation.\n");
}

FILE    *openfile(char *filename,char *mode)
{
    FILE    *file;

    file=fopen(filename,mode);
    if(file==NULL){printf(text[9],filename);exit(0);}
    printf(text[10],filename);
    return(file);
}

void    closefile(FILE *file,char *filename)
{
    fclose(file);printf(text[11],filename);
}
```

B Index file used by code

Column 1: the index name.
Columns 2-3: the central bandpass.
Columns 4-5: the left pseudocontinuum bandpass.
Columns 6-7: the right pseudocontinuum bandpass.

```
HdeltaA 4083.500 4122.250 4041.600 4079.750 4128.500 4161.000
HdeltaF 4091.000 4112.250 4057.250 4088.500 4114.750 4137.250
CN1     4142.125 4177.125 4080.125 4117.625 4244.125 4284.125
CN2     4142.125 4177.125 4083.875 4096.375 4244.125 4284.125
Ca4227  4222.250 4234.750 4211.000 4219.750 4241.000 4251.000
G4300   4281.375 4316.375 4266.375 4282.625 4318.875 4335.125
HgammaA 4319.750 4363.500 4283.500 4319.750 4367.250 4419.750
HgammaF 4331.250 4352.250 4283.500 4319.750 4354.750 4384.750
Fe4383  4369.125 4420.375 4359.125 4370.375 4442.875 4455.375
Ca4455  4452.125 4474.625 4445.875 4454.625 4477.125 4492.125
Fe4531  4514.250 4559.250 4504.250 4514.250 4560.500 4579.250
Fe4668  4634.000 4720.250 4611.500 4630.250 4742.750 4756.500
Hbeta   4847.875 4876.625 4827.875 4847.875 4876.625 4891.625
Fe5015  4977.750 5054.000 4946.500 4977.750 5054.000 5065.250
Mg1     5069.125 5134.125 4895.125 4957.625 5301.125 5366.125
Mg2     5154.125 5196.625 4895.125 4957.625 5301.125 5366.125
Mgb     5160.125 5192.625 5142.625 5161.375 5191.375 5206.375
Fe5270  5245.650 5285.650 5233.150 5248.150 5285.650 5318.150
Fe5335  5312.125 5352.125 5304.625 5315.875 5353.375 5363.375
Fe5406  5387.500 5415.000 5376.250 5387.500 5415.000 5425.000
Fe5709  5696.625 5720.375 5672.875 5696.625 5722.875 5736.625
Fe5782  5776.625 5796.625 5765.375 5775.375 5797.875 5811.625
NaD     5876.875 5909.375 5860.625 5875.625 5922.125 5948.125
TiO1    5936.625 5994.125 5816.625 5849.125 6038.625 6103.625
TiO2    6189.625 6272.125 6066.625 6141.625 6372.625 6415.125
Ca1     8483.000 8513.000 8447.500 8462.500 8842.500 8857.500
```

```
Ca2     8527.000 8557.000 8447.500 8462.500 8842.500 8857.500
Ca3     8647.000 8677.000 8447.500 8462.500 8842.500 8857.500
```