TotallySAF Short Documentation (still in edition)

S.Yahia-cherif, F.Dournac, I.Tutusaus, A.Blanchard

July 2020

contacts: syahiacherif@irap.omp.eu, fdournac@irap.omp.eu

1 Presentation

TotallySAF is a code that estimates the cosmological parameters constraints using the Fisher formalism. Initially implemented to predict the *Euclid* satellite performances (see [1], TotallySAF can compute the Fisher matrix of the photometric galaxy clustering, the spectroscopic galaxy clustering, the weak lensing and the cross-correlations between the photometric galaxy clustering and the weak lensing. TotallySAF can also combine the probes. TotallySAF has many settings that allows to estimate constraints using other surveys specifications as well. The public code is splitted up in two parts: SpecSAF a python code which estimates the constraints for the spectroscopic probe (spectroscopic galaxy clustering), and XSAF a C++ code which determine the constraints of the photometric probes (photometrix galaxy clustering, weal lensing and cross correlation). TotallySAF is a very fast code allowing to get the constraints of all probes in less than two minutes in a *Euclid* standard case. TotallySAF ensures a very good management of the memory. The code can be run on a 16Go memory device without any memory problem. Many parts of the codes are implemented in parallel using CPU. The code also contains a script allowing the user to plot contours. TotallySAF is currently available in MacOS and Linux. The code is quite dense and admits a lot of input parameters. Please read this document carefully. This document is permanently edited. The current version will only help you to launch the code, change the settings, and plot the contours. It won't iihelp you to understand all the equations and subtleties. That's why we strongly encourage the users to read [1]. Also, here are some further readings: [2, 3].

2 What you need to install

In order to run TotallySAF, you'll need the following:

- Python 2 or Python 3.
- A recent version of Scipy which uses CubicSpline interpolations. You'll also need matplotlib.
- The python module multiprocessing if you don't have it.
- A recent version of GCC (7.3.0 and higher) with a g++ compiler.
- A recent version of QT (at least 5.x.x, recommended 5.9.7) or a QMake which uses a recent version of QT. If you use Anaconda, make sure that the QT version satisfies the requirements.

3 How to run TotallySAF

3.1 First run

Here are the steps to run TotallySAF the first time:

- Open a terminal.
- Run the command: cd TotallySAF_Alpha/QTLauncher.
- Run the command: *make clean*.
- Run the command: *make -project*.
- Open *QTLauncher.pro*.
- Add on the beginning the command: *QT += widgets*.
- Run the command: *cd ../*.
- Run the command: *python launcher.py* or *python launcher_linux.py* depending on your OS.
- If it works a graphic window should display. Just chose your settings and click confirm to run the codes.

3.2 Other runs

Once the first run succeed, you can directly run the command: *python launcher_macOS.py* or *python launcher_linux.py* depending on your OS. TotallySAF hasn't been tested yet on Windows OS. The tests will be done soon.

4 Settings

Once TotallySAF runs, a window with many settings appears. The window is splitted up in 5 distinct parts: the settings of SpecSAF, the settings of XSAF, the settings of the parameters including the choice of their fiducial values and the steps of the derivatives. The two last parts concerns extra options like the cosmology, the use of a Boltzmann code, optimization options...

4.1 XSAF

- **Photometric Galaxy Clustering (GC_p):** Select yes if you want to compute the GC_p Fisher matrix. Otherwise, select no.
- **Weak Lensing (WL):** Select yes if you want to compute the WL Fisher matrix. Otherwise, select no.
- **Cross Correlations (XC: $GC_p + WL + XC$):** Select yes if you want to compute the $GC_p + WL + XC$ Fisher matrix. Otherwise, select no.
- **Number of redshift bins:** Number of tomographic redshift bins.
- **Redshift min:** Minimum redshift value.
- **Redshift max:** Maximum redshift value.
- **Number of multipoles ℓ:** Number of multipoles for which you'll compute the C_ℓ.
- **ℓ min (GC_p and WL):** Multipole minimum value for each photometric probe.
- **ℓ max (GC_p):** Multipole maximum value for the photometric galaxy clustering.
- **ℓ max (WL):** Multipole maximum value for the weak lensing.
- **Surface area ($\mathbf{deg^2}$):** Surface area of the survey.
- **Galaxy density ($\mathbf{arcmin^{-2}}$):** Total Galaxy density.

- **Number of steps (redshifts integrals):** This precision parameter concerns all the integrals computed between the minimal redshift and the maximal redshift. If you change this value, you'll have to call Camb in order to compute the new power spectrums because they're not interpolated on the redshift.
- **Use zcut option at $z > 0.9$:** This option consists in cutting the zmax value of the Galaxy clustering probe at 0.9 to make the bridge with the spectroscopic redshift range. This is an option used to compute the $GC_p + GC_s + WL + XC$ pessimistic Fisher matrix. In other words we consider the galaxy clustering probe won't detect any galaxy in the spectroscopic range. The photometric bias values at $z > 0.9$ will be then fixed to 0.
- **Observed ellipticities variance:** This is the Observed ellipticities variance for the weak lensing probe.
- **Compute Photo-z:** If you already have a photo-z table, you won't need to compute it again. Keep in mind that each time you change the number of redshift bins, the redshift range, the photo-z model or the galaxy density distribution, you'll have to compute the new photo-z.
- $\boldsymbol{z_0}$**:** galaxy density distribution $z_0 = 0.9/\sqrt{2}$.
- $\boldsymbol{\alpha}$**:** galaxy density distribution exponential index.
- $\boldsymbol{c_b}$**:** Multiplicative bias for the well determined redshift sources.
- $\boldsymbol{z_b}$**:** Additive bias for the well determined redshift sources.
- $\boldsymbol{\sigma_b}$**:** Photometric redshift precision. $\sigma_b(1+z)$ for the well determined redshift sources.
- $\boldsymbol{c_o}$**:** Multiplicative bias for the catastrophic redshift sources.
- $\boldsymbol{z_o}$**:** Additive bias for the catastrophic redshift sources.
- $\boldsymbol{\sigma_o}$**:** Photometric redshift precision for the catastrophic redshift sources.
- **Fraction Outliers:** This is the fraction of catastrophic redshift measurements.

4.2 SpecSAF

- **Number of redshift bins:** Total number of redshift bins. Note that the spectroscopic probe isn't very sentitive to the number of redshift bins as long as the redshift size of each bin is lower than 0.25.
- **Redshift min:** Minimum redshift bin value.
- **Redshift max:** Maximum redshift bin value.
- **Mean redshift bin:** Mean redshift bin (the redshift in which the linear galaxy velocity dispersion σ_p and σ_v will be estimated).
- **Min scale (h/Mpc):** Minimum scale value.
- **Max scale (h/Mpc):** Maximum scale value.
- **Linear or non linear $\boldsymbol{P(k)}$:** For now the non linear option isn't available as SpecSAF doesn't compute the nonlinear matter power spectrum. This feature will come soon. However a SNL (semi-non linear) option is available. This is the one used by the IST in which the Redshift space distortion (RSD) term takes the non linear form.
- **Power spectrum case:** If you choose the semi-non linear case, you can switch between two options. The default one used by the IST is the option SNL2.
- **CMB temperature (NW $\boldsymbol{P(k)}$):** This is the CMB temperature used to compute the no wiggle Power spectrum. Make sure it's value corresponds to the one you use in Camb.

- **Number of massive neutrinos (NW $P(k)$):** This is the number of massive neutrinos to compute the no wiggle power spectrums. Make sure it's value corresponds with the one you use in Camb.
- **ODE or fit projection:** During the projection part, the growth rate $f(z)$ can be solved by computing an ordinary differential equation (more precise) or by using the fit: $f(z) = \Omega_m(z)^\gamma$ (less precise). We highly recommend to always use the ODE option if you don't estimate the constraints over γ. However you'll have to use the fit formula if you want to estimate the γ parameter constraints.
- **Surface area (deg^2):** Surface area of the survey.
- **Spectroscopic precision:** Spectroscopic precision value.
- **Derivatives scheme:** The derivative scheme used is the centered scheme. You can choose between 3, 5 and 7 points. The 3 points is faster but less precise whereas the 7 points is the most precise one but the slowest one.
- **Integrals number of steps:** This is a precision parameter of the double integrals needed to compute the intermediate Fisher matrix before projection.
- **Steps for the projection:** This is the step size used for the projection part. The method used is the 3 points scheme. The steps will be the same for all the parameters because the method is very stable numerically.
- **ODE number of steps:** Number of points to solve the ODE. The more this value increase the more precise is the method. However computing time will also increase
- **Use $b(z)$ or $\ln b\sigma_8(z)$:** Initially, the IST use the $\ln b\sigma_8(z)$ parameter (bias at a given redshift times $\sigma 8(z)$) which is marginalized before the projection. SpeCSAF don't make any marginalization before projection to keep all the parameters on the Final Fisher matrix. Both methods still give the same results. However avoiding marginalization allow to plot all the parameters contours. This won't be possible anymore if some parameters are marginalized before the projection. Here we allow the user to keep $\ln b\sigma_8(z)$ or to project into $b(z)$. Note that choosing one or the other won't change the constraints over the cosmological parameters or the shot noise.

4.3 Parameters

The parameters settings section is separated in 4 columns:

- **Parameters:** This is the name of the parameters.
- **Use:** If this option is set to yes, the corresponding parameter will be a free parameter. If you select no the corresponding parameter will be fixed.
- **Fiducials:** This is the fiducial value of the parameter.
- **Steps:** This is the step size of the derivative.

All the parameters are presented here:

- **Ω_m:** Matter physical density
- **Ω_{DE}:** Dark energy content directly linked to the curvature of the universe ($\Omega_{DE} = 1 - \Omega_{DE}$ if flat universe).
- **Ω_b:** Baryon density parameter.
- **w_0:** Dark energy equation of state at $z = 0$.
- **w_a:** Dark energy derivative with respect to the scale factor.

- $\boldsymbol{h}$**:** Reduced Hubble constant defined as $h = H_0/(100\text{km.s}^{-1}.\text{Mpc}^{-1})$ where H_0 is the present day Hubble parameter.
- $\boldsymbol{n_s}$**:** Spectral scalar index.
- $\boldsymbol{\sigma_8}$**:** The root mean square of present-day linearly evolved density fluctuations in spheres of 8 h^{-1}Mpc.
- $\boldsymbol{\gamma}$**:** Growth index parameter.
- $\boldsymbol{\sigma_p}$**:** Linear galaxy velocity dispersion (1st parameter).
- $\boldsymbol{\sigma_v}$**:** Linear galaxy velocity dispersion (2nd parameter).
- $\boldsymbol{GC_s}$ **bias:** Spectroscopic bias.
- $\boldsymbol{GC_s}$ **shot noise:** Spectroscopic shot noise.
- $\boldsymbol{A_{IA}}$ **(WL):** Intrinsic alignment (eNLA) model (1st parameter).
- $\boldsymbol{B_{IA}}$ **(WL):** Intrinsic alignment (eNLA) model (2nd parameter).
- $\boldsymbol{n_{IA}}$ **(WL):** Intrinsic alignment (eNLA) model (3rd parameter).
- $\boldsymbol{C_{IA}}$ **(WL):** Intrinsic alignment (eNLA) model (4th parameter).
- $\boldsymbol{GC_p}$ **bias:** Photometric bias.
- $\boldsymbol{\ln[D_a(z)](GC_s)}$**:** Angular distance at a given redshift bin.
- $\boldsymbol{\ln[H(z)](GC_s)}$**:** Hubble parameter at a given redshift bin.
- $\boldsymbol{\ln[f\sigma_8(z)](GC_s)}$**:** Growth rate at a given redshift bin times $\sigma_8(z)$.
- $\boldsymbol{\Omega_\nu}$**:** Neutrinos density parameter.
- $\boldsymbol{A_s(\times 10^{-9})}$**:** Amplitude of the scalar mode.

4.4 Other options

- **Use XSAF: Select yes to compute the Fisher matrix for the photometric probes. Select no if you only want to compute the Fisher matrix of the spectroscopic probe.**
- **Use SpecSAF:** Select yes to compute the Fisher matrix for the spectroscopic probe. Select no if you only want to compute the Fisher matrix of the photometric probes.
- **Flat or non flat case:** For the flat cosmology, the use Ω_{DE} option has to be set to no. In this case, $\Omega_{DE} = 1 - \Omega_m$. If you select non flat case, don't forget to set yes concerning the use Ω_{DE} option.
- **Use Camb:** If you don't have the power spectrums for the case you want to study, you'll have to use Camb to compute the matter power spectrums. That's to say, each time you change the derivatives steps, the redshift range, the number of redshift bins, the Number of steps for the integrals (XSAF), you'll have to compute the matter power spectrum.
- $\boldsymbol{\ell}$ **cut optimization:** This option is a feature of XSAF. After computing the C_ℓ, the covariance matrix of the observables is built. Generally the size of the covariance matrix produced is enough high to ensure a slow inversion procedure. The ℓ cut optimization consists in building sub Fisher matrix for one or multiple ℓ values. On the end of the process, we compute the sum of each Fisher matrix to get the final Fisher matrix. To do so, we cut the covariance matrix into small independent blocks. Each small block is individually inverted. As the time to invert a matrix doesn't evolve linearly with the size of the matrix, inverting multiple small covariance matrix is much faster than inverting one big covariance matrix. As an example, let's consider the following situation: we want to compute a Fisher matrix for XSAF with all the probes with the same ℓ_{max} value. We take 10 tomographic bins and 60

multipoles. The covariance matrix size is $N_{multipoles}(NT_{GC_p} + NT_{WL} + N_{XC})$. With $N_{multipoles}$ the number of multipoles (here 60), NT_{GC_p} and NT_{WL} all the unique pairs of tomographic elements (both 55) and N_{XC} all the pairs of tomographic elements (here 100). The covariance matrix size will be $60(2*55+100) \times 60(2*55+100) = 12600 \times 12600$. The inverting procedure last around three minutes. Now let's consider you decided to use the ℓ cut option with 30 values. As the total number of multipoles is 60, the number of multipoles selected per cut will be 2. In other words, each sub covariance size will become $2(2*55+100) \times 2(2*55+100) = 420 \times 420$. Inverting such small matrix last less than one second. In other words you'll win 2 minutes and 30 seconds on the total computation time which represents a lot as the total computation time is around 2 minutes with the ℓ cut option presented above. Moreover this feature is very interesting for huge covariance matrix. Generally computing large covariance matrix isn't memory friendly. But with the ℓ option, you can easily build a matrix with several dozens of tomographic bins without memory issues. Note that if you choose 1 as a value, the cut won't be made. Use 1 if you know that your covariance matrix is very small.

- **Combine the probes:** XSAF and SpecSAF compute the GC_p, GC_s, WL and $GC_p + WL + XC$ Fisher matrix. If you use the probe combination, the following additional matrix will also be computed. $GC_p + GC_s$, $GC_p + WL$, $GC_s + WL$, $GC_p + GC_s + WL$, $GC_p + GC_s + WL + XC$. Note that combining the probe is a very fast operation. Don't hesitate to use it.

5 How to plot some contours

In order to plot the contours, you'll have to go in the Triplot directory. Here are the steps:

- First you'll have to copy the Fisher matrix in the *F_input* directory. The Fisher matrix need to have the same size with the same parameters. If you want to compare two different matrix with different size, you'll have to marginalise them.
- Open the *init_CTR.txt* file and edit it. The file is separated up in three parts. The first part concern the cosmetic aspects of the code, like the size of the ticks, the size of the labels, the legend background color... The second part concerns the parameters. You'll have to enter the parameter name using *LaTeX* notations, add a space, and write the fiducial value. They have to follow the order of the Fisher matrix. The last part concerns the legend. You have to write as many entries as the Fisher matrix in your *F_input directory*. In order to display the legend in the correct order, you have to write it in the format: File name in the *F_input directory* | legend name. Only the legend name is displayed on the code.
- Open the colors file and edit it if you want to add new colors or change the colors order. If this file contains more entries than Fisher matrix, the code will take the first entries. Note that this file has to contain at least as many entries than the number of Fisher matrix.
- Run the code: *python triplot_CTR.py*. The contours are saved in pdf and png format. You can find them in the *tri_output* directory.

6 Next features and improvements

The code can still be improved. The planned features are the following:

- The galaxy density distribution is currently common between the photometric galaxy clustering and the weak lensing. We plan to make these two distributions as distinct ones.
- The choice of the non linear model in Camb cannot be selected on the graphical interface. By default, XSAF uses Takahashi from Camb. We will add the choice of the non linear model directly on the graphical interface. We'll also let the user to choose the linear model.
- SpecSAF doesn't use nonlinear model from the Boltzmann codes like XSAF. We plan to include the non linear models to XSAF.

- CLASS can be used with TotallySAF. However the current public version of TotallySAF only uses Camb. CLASS will be added on the public version.
- The checking of inconsistencies on the settings hasn't been implemented yet for the XSAF parameters. This will be done soon.
- The order of the parameters isn't displayed in the Fisher matrix. The order is the same as presented in section 4.3. Only consider those which have a use option. For the redshift dependent parameters, each same parameters is written next to each other with the redshift increasing. Example for the bias and the shot noise: $b(z_1)$, $b(z_2)$, ... $b(z_N)$, $PS(z_1)$, ... $PS(z_N)$...
- The graphic window doesn't display any information about the settings if you put your cursor on it. For now, refer to this document to know what are all the parameters.
- XSAF was initially implemented in Python. In order to considerably reduce the running time, the public version of XSAF has been implemented in C++. The Python version of XSAF will be added.
- Initially the SpecSAF code was computing the derivatives until the 15 points stencil. Stability and optimization tests have shown that the 3 points stencil is more than enough to get stable results. Moreover this is the fastest scheme as it asks less operations. That's why SpecSAF is restrained to the 3, 5, or 7 points stencil. However, if requested, higher stencils can be added to the codes. XSAF is currently using the 3 points stencil only.
- MCMC implementation: a MCMC version of SpecSAF already exists. However it's not yet added on the public version.
- TotallySAF hasn't been tested yet using a Windows OS. There's no guarantee the code works for Windows users.

Keep in mind that the new features will come or not depending on the developers time.

7 Known issues

The code is actually on an Alpha version which means some bugs or glitches can still exist. If you find or suspect any issue, please inform the contacts listed above.

References

[1] Collaboration, Euclid, A. Blanchard, S. Camera, C. Carbone et al.: *Euclid preparation: VII. Forecast validation for Euclid cosmological probes*. 2019.

[2] Tutusaus, I., M. Martinelli, V.F. Cardone et al.: *Euclid: The importance of galaxy clustering and weak lensing cross-correlations within the photometric Euclid survey*. 2020.

[3] Yahia-Cherif, S., A. Blanchard, S. Camera et al.: *Validating the Fisher approach for stage IV spectroscopic surveys*. 2020.