Users' Guide for CSWAB 1.0

Hu Zhan
Department of Physics, University of California, Davis, CA 95616, USA
zhan@physics.ucdavis.edu

August 28, 2006

Contents

1 Introduction

CSWAB utilizes the Fisher information matrix to forecast constraints on cosmological (and other) parameters from the **C**osmic microwave background (CMB), **S**upernova, **W**eak lensing (WL), **A**nd **B**aryon acoustic oscillation (BAO) data. I assume a photometric redshift (photo-z) survey for BAO and WL. Spectroscopic BAO and supernova components will be available in the future.

I implemented CSWAB for the calculations in Zhan (2006). This code is primarily written in `C++` in a quasi-object-oriented fashion, so that it may be easily adapted for other applications.

CSWAB 1.0 contains 4 packages: auxiliary functions, evolution of cosmological quantities (including cosmological parameters, distance, linear growth function, etc.), CMB, and photometric BAO/WL, with each one depending on the preceding one(s).

CSWAB takes advantage of the GNU Scientific Library[1] (GSL, version 1.6) for numerical algorithms and relies on CMBFAST version 4.5.1[2] (Zaldarriaga & Seljak 2000) for CMB power spectra and transfer functions. It is parallelized through the massage passing interface[3] (MPI, standard 1.1). Although one has the option to compile and run CSWAB as a serial application, it may take too long (e.g., days) and/or too much memory to carry out a production run of, say, 50 galaxy and shear bins on a single computer.

Cosmological parameters forecast in CSWAB 1.0 include the dark energy equation of state (EOS) parameters w_0 and w_a, where the EOS is parametrized as $w(a) = w_0 + w_\mathrm{a}(1-a)$, the matter density ω_m, the baryon density ω_b, the angular size of the sound horizon at the last scattering surface θ_s, the equivalent matter fraction of curvature Ω_K, the optical depth τ to scattering by electrons in the reionized intergalactic medium, the primordial helium mass fraction Y_p, the spectral index n_s of the primordial scalar perturbation power spectrum, the running of the spectral index α, and the normalization of the primordial curvature power spectrum Δ^2_R at $k = 0.05\,\mathrm{Mpc}^{-1}$. I adopt the three-year *WMAP* results Spergel et al. (2006) for fiducial values of the parameters: (w_0, w_a, ω_m, ω_b, θ_s, Ω_K, τ, Y_p, n_s, α, Δ^2_R) = (-1, 0, 0.127, 0.0223, 0.596°, 0, 0.09, 0.24, 0.951, 0, 2.0×10^{-9}). The reduced Hubble constant $h = 0.73$ and the present equivalent matter fraction of dark energy $\Omega_\mathrm{X} = 0.76$ are implicit in my parametrization.

I assume in this guide that you have unpacked `cswab.tgz` and that the current directory is the CSWAB home directory: `some_directory/cswab/`. File names, directory names, commands, and file contents are in typewriter font. Optional arguments are enclosed in square brackets.

Bugs/Questions: Email zhan@physics.ucdavis.edu.

Disclaimer: CSWAB is provided as is and without warranties of any kind. The user must agree to use CSWAB at his/her own risk.

2 Limitation

CSWAB 1.0 is based on very ideal assumptions. For example, I have not considered non-Gaussian photo-z errors, or different galaxy bias and photo-z properties associated with galaxy types (e.g., Schneider et al. 2006), or more complex behavior of the systematics (e.g., Huterer et al. 2006).

I should also mention that the transformation between the linear matter power spectrum to the nonlinear one in CSWAB is *ad hoc*. Although the Peacock & Dodds (1996) fitting formula works well for a no-wiggle power spectrum, it does not make sense for a "real" power spectrum, in which BAOs are present. The reasons are that it shifts the scales of BAO features unphysically:

$$k_\mathrm{L} = [1 + \Delta^2_\mathrm{NL}(k_\mathrm{NL})]^{-1/3} k_\mathrm{NL}, \tag{1}$$

$$\Delta^2_\mathrm{NL}(k_\mathrm{NL}) = f_\mathrm{NL}[\Delta^2_\mathrm{L}(k_\mathrm{L})], \tag{2}$$

where $f_\mathrm{NL}(x)$ is determined by the spectral index $n_\mathrm{L}(k_\mathrm{L}) = \mathrm{d}\ln P/\mathrm{d}\ln k|_{k=k_\mathrm{L}/2}$ (and the growth function), and that the mapping can be non-monotonic due to the oscillations in n_L. Other fitting formulae that depend on the spectral index may face the same problem. My solution is to use a

[1] See http://www.gnu.org/software/gsl/.
[2] See http://www.cmbfast.org/.
[3] See http://www-unix.mcs.anl.gov/mpi/.

no-wiggle power spectrum $\Delta^2_{\rm NW,L}(k)$ (Eisenstein & Hu 1999) that otherwise matches the real power spectrum $\Delta^2_{\rm L}(k)$ with BAOs to obtain the boost $\Delta^2_{\rm NW,NL}(k)/\Delta^2_{\rm NW,L}(k)$ and then apply it to $\Delta^2_{\rm L}(k)$.

Alternatively, one may obtain the nonlinear power spectrum with BAOs using the halo model (e.g., Cooray & Sheth 2002), perturbation theory (Jain & Bertschinger 1994; Jeong & Komatsu 2006), and simulations (Seo & Eisenstein 2005; Springel et al. 2005; White 2005). These methods all have their own limitations: the halo model is less accurate in the quasi-linear regime (the computation is more intensive as well), perturbative calculations diverge at $z \lesssim 1$, and simulations are time consuming (ultimately, the nonlinear power spectrum should be calibrated by simulations).

Given my treatment of the nonlinear power spectrum and the fact that I do not include neutrinos, results of the parameters related to the power spectral index, such as the initial tilt $n_{\rm s}$ and running α, may be less robust.

3 Building CSWAB

Before compiling CSWAB, you need to set up the dependencies: MPI (needed only if you want a parallel CSWAB), GSL, and CMBFAST version 4.5.1. Follow the standard procedures to install MPI (I use `mpich` and have not tested others) and GSL. Specific modifications to CMBFAST and instructions for CSWAB are described as follows.

3.1 Interfacing with CMBFAST

One powerful CMB constraint is the angular size of the sound horizon at the last scattering surface, $\theta_{\rm s}(w_0, w_{\rm a}, \omega_{\rm m}, \omega_{\rm b}, Y_{\rm p}, \Omega_{\rm K}, \Omega_{\rm X})$. It is a useful and important parameter to include for forecasting CMB constraints or incorporating CMB priors with forecasts of other experiments.

In order to take numerical derivatives of the power spectra with respect to other parameters at a fixed $\theta_{\rm s}$, one must trade the implicit parameter $\Omega_{\rm X}$ with the others. Fitting formulae are convenient for approximating $\theta_{\rm s}$ to the percent level, but given that $\theta_{\rm s}$ can be so precisely measured, e.g., $\sigma(\theta_{\rm s}) \sim 0.0002^\circ$ from *Planck*, these formulae may not be accurate enough for determining the exact trade-offs between $\Omega_{\rm X}$ and the other parameters.

To build a consistent forecasting tool, I extract $\theta_{\rm s}$ from CMBFAST, which supplies the distance to the peak of the visibility function (occurred at `armax`), `dlsso`, and the sound horizon at the peak, `shor`, in `subroutine cmbflat` (`cmbflat.F`) and `subroutine cmbopen` (`cmbopen.F`). Hereafter, I only cite the flat case for brevity. A naive extraction of $\theta_{\rm s}$ may lead to errors as large as $\Delta\theta_{\rm s} \sim 0.0001^\circ$. *This does not affect CMBFAST results*, because `dlsso`, `shor`, and `armax` are not used in any way that can impact the computation in CMBFAST appreciably. Rather, the errors will affect the calculations of the input parameters of CMBFAST and lead to incorrect numerical derivatives.

The error of the extracted $\theta_{\rm s}$ is primarily caused by the fact that the extremum is searched over arrays in `subroutine finithermo` (`cmbflat.F`, line 3067) without interpolation. Other factors exacerbate the problem, but their own contribution to the error – when `armax` is interpolated – is $\lesssim 10^{-5}$ deg. For instance, one can get as much as $\sim 0.0001^\circ$ differences in $\theta_{\rm s}$ by assigning different values to the maximum wavenumbers `akmaxt` for transfer functions or `akmax0` for CMB. The reason is that both `akmaxt` and `akmax0` affect the starting point, `taumin`, for initializing the thermal/ionization history of baryons in `subroutine finithermo` and, hence, the peak can shift from one place in the arrays to another. Interested readers may want to check the value of `shor`/`dlsso` in the original CMBFAST as `akmaxt` varies (try `cmbfast_directory/cmb < src/cmb/k1.par` and `cmbfast_directory/cmb < src/cmb/k100.par`).

By the same token, the specified fiducial value of $\theta_{\rm s}$ is not precise enough. Therefore, the actual fiducial value is calculated from the other parameters.

I list the essential modifications in this subsection; for others, see the results of `diff` between the modified and original CMBFAST codes in `src/cmb/cmbfast.dif`. After finishing the modifications, compile CMBFAST with `HP` and `UNNORM` options. Do not use `MPIBUILD`.

3.1.1 Increasing the Resolution in `cmbfast.inc` and `subroutines.F`

Modifications are made to increase the resolution in general. For `cmbfast.inc`:

1. *Position* Line 29
   Original `parameter(nthermo=10000)`
   Modified `parameter(nthermo=40000)`
   Remarks To increase `nthermo` above 40000, one must increase the array sizes in `subroutine splder` and `subroutine splini`, which are called by `subroutine finithermo`.

2. *Position* Line 38
   Original `parameter (nk0=300)`
   Modified `parameter (nk0=1200)`
   Remarks To get smooth BAOs, `src/cmb/cmb.h` sets the output frequency, `NK_PER_DEX`, of the transfer function to 150 per dex of wavenumber. CMBFAST starts output at $k \sim 10^{-5}\ h\,{\rm Mpc}^{-1}$. To get $k \sim 10\ h\,{\rm Mpc}^{-1}$ for WL, you need `nk0` $\gtrsim 900$.

3. *Position* Line 40
   Original `parameter (nstep0=7000)`
   Modified `parameter (nstep0=15000)`
   Remarks Required for the concordance model and small perturbations around it.

Modifications to `subroutines.F`:

1. *Position* Line 1023 & 1025, `subroutine output_power`
   Original `parameter(nmax=1000)`
   `parameter(kmax=1000)`
   Modified `parameter(nmax=nk0)`
   `parameter(kmax=nk0)`
   Remarks To avoid indexing beyond the array bound.

3.1.2 Extracting the Sound Horizon from `cmbflat.F` and `cmbopen.F`

The following code fragment from `cmbflat.F` (line 363)

```
call finithermo(taumin,taumax,tau0,taurend,dlntau0,n1,nstep)
dlsso=tau0-taurmax
shor=rombint(dsoundda,1.0d-8,armax,tol)
```

gives the distance to the peak of the visibility function (occurred at the conformal time `taurmax` or the expansion factor `armax`), `dlsso`, and the sound horizon at `armax`, `shor`. As mentioned above, the extremum is searched over arrays without interpolation in `subroutine finithermo` (line 3067):

```
vismax=0.0d0
do i=1,nthermo
   tvis=dotmu(i)*emmu(i)
   if (tvis.gt.vismax) then
      vismax=tvis
      tau=tauminn*exp((i-1)*dlntau)
      taurmax=tau
      armax=atemp(i)
   endif
end do
```

Consequently, the numerical derivatives can be erratic. To alleviate this problem, I supplement the search with an interpolation to the point where the first derivative of the visibility function vanishes:

```
vismax=0.0d0
do i=1,nthermo
   tvis=dotmu(i)*emmu(i)
   if (tvis.gt.vismax) then
      vismax=tvis
      tau=tauminn*exp((i-1)*dlntau)
      taurmax=tau
      armax=atemp(i)
      nvmx1 = i
   endif
end do

dv1 = ddotmu(nvmx1) * emmu(nvmx1) + dotmu(nvmx1) * demmu(nvmx1)
if (dv1 .ge. 0.d0) then
   nvmx2 = nvmx1 + 1
else
   nvmx2 = nvmx1 - 1
endif
dv2 = ddotmu(nvmx2) * emmu(nvmx2) + dotmu(nvmx2) * demmu(nvmx2)
i = nvmx2 - 1
taurmax = (tau * dv2 - tauminn * exp(i * dlntau) * dv1)
&     / (dv2 - dv1)
armax = (armax * dv2 - atemp(nvmx2) * dv1) / (dv2 - dv1)
```

Now θ_s responds more smoothly to the parameters. Similar changes to `cmbopen.F` are documented in `src/cmb/cmbfast.dif`.

3.1.3 Curvature in `recfast.f`

Finally, there is a minor bug in `subroutine recfast(OmegaB,OmegaC,omegav,HOinp,tcmb,Yp, annur)` (`recfast.f`, line 256):

```
OmegaT=OmegaC+OmegaB            !total dark matter + baryons
OmegaK=1.d0-OmegaT-OmegaL       !curvature
Omegal=omegav
```

where the last two lines should have been switched.

Since `OmegaL` (case insensitive) is never assigned before being used, the curvature term `OmegaK` is indeterminant. Fortran compilers may set all automatic variables to 0 *once* when the program starts (*not* every time the subroutine is called), so that one gets `OmegaK=1.d0-OmegaT` regardless the input value of `omegav`. If the subroutine is called a second time, then `OmegaL` may retain its value from the first call until being reassigned (again, depending on the behavior of the compiler), which could result in a correct value of `OmegaK` if there were no changes in `OmegaB`, `OmegaC`, and `omegav` between the two calls. The impact of this bug alone on $\theta_{\rm s}$ is $\lesssim 10^{-5}$ deg, because `OmegaL` ~ 0 to high degree at recombination.

3.2 Building CSWAB: A "Quick" Start

First, you need to set several path names in `src/Makefile`:

```
JOBDIR  = directory_of_cswab
CMBFDIR = directory_of_cmbfast
SYSINCDIR = directory_of_GSL_and_MPI_headers
SYSLIBDIR = directory_of_GSL_and_MPI_libraries
```

The last two variables may be left empty if GSL and MPI can be found in the default system header and library directories. Then, choose `C++` and Fortran compilers and set their switches:

```
CXX = mpiCC
CXXFLAGS = -O3 -Wall -Wno-deprecated
FC  = g77
FFLAGS = -O3 -DHP -DUNNORM -I$(CMBFDIR)
```

Here `mpiCC` is an MPI wrapper of `g++` on my system. Finally, set the MPI option and variables (choose a parallel `C++` compiler above as well) if you want a parallel CSWAB:

```
OPTIONS += -DMPI_CODE
MPIRUN = /usr/bin/mpirun -np 2
MPILIB = mpich
```

Otherwise, these entries should be left empty.

Now, type `make` in the CSWAB home directory or in the source directory `src/`. If everything goes well, three executables: `cmbfm`, `wlbao`, and `clcon` will be generated in `bin/` in a short while.

3.3 Tests

Two sets of tests are available for *Planck* and the Large Synoptic Survey Telescope[4] (LSST) with 10 galaxy bins and 5 shear bins. For these tests, do not change any `src/Makefile` variables or options except those mentioned above.

The first one, `example`, requires you to download the pre-calculated CMB power spectra, transfer functions, and BAO/WL power spectra (90MB total). Once the files are in place (`example/`), you can run it by issuing the command `make example` in the CSWAB home directory. The results should be approximately the same as those in `example/log`.

The second one, `test`, calls CMBFAST to compute the CMB power spectra and transfer functions, calculates the galaxy/shear power spectra from scratch, and forecasts the errors. To run this

[4]See http://www.lsst.org/.

one, issue the command `make test` in the CSWAB home directory. It takes about 5 hours to finish on a dual Opteron 1.8 GHz. The results should be similar to those in `test/log`.

3.4 More Options

You may turn on a number of options in `src/Makefile` with `OPTIONS += -Doption_name`[=*value*]. These options are grouped in the following categories.

1. CMB experiment: `CMBPOL`, `PLANCK`, & `WMAP`

   Choose *one* of the options for the noise characteristics of the CMB experiment, which are hardwired in `src/cmb/cl.cpp`. You have to specify the correct sky fraction at the command line and, if desired, modify the ℓ ranges in `src/cmb/cmb.h`.

2. Debugging: `VERBOSE` & `OUT_CL`

   Setting `VERBOSE=0` (or not switching it on at all) leads to no extra output; `1`, `2`, and `3` produce progressively more verbose logs. Set `OUT_CL=`n, where n is an integer, to output $P_{ij}(\ell)$ and the derivatives at $\ell = n$.

3. Power Spectrum: `LINEAR_PS` & `NINJCL`

   Turn on `LINEAR_PS` to compute the galaxy and shear power spectra using the linear matter power spectrum. `NINJCL` enables one to calculate the Fisher matrix with $\bar{n}_i\bar{n}_j P_{ij}(\ell)$ instead of $P_{ij}(\ell)$ (Ma, Hu, & Huterer 2006). Note that `NINJCL` is implemented only for `DA_F_DB`.

4. Fisher Matrix Algorithm: `DA_F_DB` (recommended)

   For zero-mean observables (e.g., the map data in Fourier space) with no scale-scale correlations, one may calculate the Fisher matrix in two ways (Hu & Jain 2004):

   $$F_{\alpha\beta} = f_{\rm sky} \sum_\ell \frac{2\ell+1}{2} \,{\rm Tr}\, \boldsymbol{C}_\ell^{-1} \frac{\partial \boldsymbol{C}_\ell}{\partial p_\alpha} \boldsymbol{C}_\ell^{-1} \frac{\partial \boldsymbol{C}_\ell}{\partial p_\beta} \tag{3}$$

   $$= f_{\rm sky} \sum_\ell (2\ell+1) \frac{\partial \boldsymbol{Q}_\ell^{\rm T}}{\partial p_\alpha} \boldsymbol{M}_\ell^{-1} \frac{\partial \boldsymbol{Q}_\ell}{\partial p_\beta}, \tag{4}$$

   where p_α is a parameter, $\boldsymbol{C}_\ell$ is the covariance, i.e., the auto and cross power spectra, of the observables in Fourier space, $\boldsymbol{Q}_\ell$ is a column vector of unique elements of $\boldsymbol{C}_\ell$, and $\boldsymbol{M}_\ell$ is the covariance of $\boldsymbol{Q}_\ell$. $\boldsymbol{M}_\ell$ can be constructed analytically with the elements of $\boldsymbol{C}_\ell$.

   Equation (3) is easy to implement but inefficient and sometimes unstable. For example, although $\boldsymbol{C}_\ell^{-1}$ is supposed to be symmetric, transposing it once in the equation (or just taking "advantage" of the symmetric matrix product in `BLAS`) may lead to large errors. It is recommended to polish the inverse $\boldsymbol{C}_\ell^{-1}$ a few times for equation (3).

   The CMB package always uses the second method, equation (4); for BAO and WL, you can enforce it with `DA_F_DB`. Actually, it must be turned on if one wishes to forecast with pre-calculated galaxy and shear power spectra (see `src/gals/ctom.h` & `ctom.cpp`).

5. Growth Function for WL: `WL_NO_G`

   This option is devised to investigate the origin of cosmological constraints from WL. `WL_NO_G` assigns artificial bias parameters [fiducial value $b(z) = 1$] to WL shear power spectra (Knox,

Song, & Zhan 2006). It not only limits WL's ability to measure the amplitude (or growth rate) of the matter power spectrum, but also degrades WL's distance measurements.

6. Redshift Range of Parameters: `EXTEND_PARS` (recommended)

   The galaxy bias and photo-z parameters are assigned in true-redshift space; they have nothing to do with the binning of galaxies in photo-z space. Without `EXTEND_PARS`, the parameters are spaced between $z = 0$ to (the numerical value of) the maximum photo-z $z_{\rm p,max}$ = `Z_MAX` (`src/gals/swab.h` & `swab.cpp`) of the survey. Because of photo-z errors, the galaxy distribution in true-redshift space spreads beyond z = `Z_MAX`. Therefore, it is reasonable to extend the parameters to z = `Z_MAX` + `Z_EXT` with the option `EXTEND_PARS`, where `Z_EXT` = 0.5 can be adjusted in `src/gals/swab.cpp`. This extension does not add more parameters but only increase the redshift interval between parameters.

4 Running CSWAB

There are three driver programs `cmbfm.cpp`, `wlbao.cpp`, and `clcon.cpp` in `src/`. They provide simple examples of utilizing CSWAB classes and are flexible enough for a variety of applications. For brevity, I define a boolean argument `x` to be *true* if and only if `x` can be interpreted as a non-zero integer or a string starting with `t` or `y` (case insensitive).

4.1 CMB

Program `bin/cmbfm`

Purpose To generate a CMB Fisher matrix.

Syntax `cmbfm f_sky Cl_dir f_name` [`cal_Cls cos_par cmbfast`]

- `f_sky` Effective sky fraction covered by the experiment.
- `Cl_dir` Directory to write and/or read CMB power spectra and transfer functions.
- `f_name` File name of the output Fisher matrix .
- `cal_Cls` Calculate the CMB power spectra and transfer functions if `cal_Cls` is *true* (default *false*).
- `cos_par` Cosmological parameter file. If not specified, the program assumes the concordance model in `src/cmb/cmb.cpp`.
- `cmbfast` CMBFAST executable name (default `CMBF_EXEC`, see `src/Makefile`).

Remarks Pre-calculated CMB power spectra must exist in the directory `Cl_dir` if `cal_Cls` is *false*. Otherwise, `Cl_dir` must be writable.

Here is an excerpt from the cosmological parameter file `test/lcdm.par`:

```
% comments start with %, #, or !
w0  -1  2e-2  2
wa   0  4e-2  2
...
```

The columns are the parameter name (string), its fiducial value (float), the step size for numerical differentiation (float), and the method of differentiation (integer, always two-sided). The parameter names are stored in `COS_PAR_NAME` (`src/cmb.cpp`). The order of

the parameters is not important. The fiducial value of $\theta_{\rm s}$ will be calculated from other parameters, so that you need to specify either `Ox` for $\Omega_{\rm X}$ or `H0` for H_0. Only the fiducial value is relevant to `Ox` and `H0`. If both are given, `Ox` overrides `H0`. Entries that do not match the defined names will be discarded. If multiple entries of the same parameter are present, the last one will be in effect. Parameters that are absent in the file will assume the values in `src/cmb/cmb.cpp`.

The order of parameters in the CMB Fisher matrix follows: w_0, $w_{\rm a}$, $\omega_{\rm m}$, $\omega_{\rm b}$, reserved, $\theta_{\rm s}$, $\Omega_{\rm K}$, τ, $Y_{\rm p}$, $n_{\rm s}$, α, reserved, and Δ^2_R.

4.2 BAO and WL

Program `bin/wlbao`

Purpose To generate galaxy and shear power spectra.

Syntax `wlbao parFile n_BAO n_WL n_phz sig_z0` [`dndz_z0 N_gal cal_fsh f_name`]

`parFile` Name of the BAO/WL parameter file. Here is an example:

```
20000                    % SURVEY_AREA (deg^2)
3000    2000             % SWB_L_MAX  WL_L_MAX
0       3.5              % DNDZ_MD  Z_MAX
z       u       u        % BAO_BIN_FLG WL_BIN_FLG PHZ_BIN_FLG
0.18    0.042            % WL_RMS_SHEAR WL_DRMSS_DZ
0       0                % DUST_NOISE SHEAR_NOISE
test/cmb/transf/tf       % TFFN
test/lcdm.par            % COSFN
test/wlbao               % OUT_DIR
2       wbcl/            % OUT_CL_FLG CLFN
0                        % not used
```

`DNDZ_MD` sets $n(z)$ type: `0` for $n(z) \propto z^2 e^{-z/z_0}$, `1` for $n(z) \propto z^2 e^{-z^2/z_0^2}$, and `2` for the one in Song & Knox (2004). `*_BIN_FLG` sets the arrangement of bin widths or parameter intervals: `z` for width/interval $\propto 1+z_{\rm p}$ or `u` for uniform. The galaxy rms shear $\gamma_{\rm rms}$ = `WL_RMS_SHEAR`+`WL_DRMSS_DZ`$\times z$. `TFFN` is the prefix of the transfer function names. `COSFN` is the name of the cosmological parameter file, and `OUT_DIR` is the output directory. `OUT_CL_FLG` sets the output mode: `0` for no output, `1` for $P_{ij}(\ell)$ in text files, or `2` for $P_{ij}(\ell)$ and derivatives in binary files. In the last two cases, the galaxy/shear power spectra will be written with the prefix `CLFN` in the directory `OUT_DIR`. See `test/wlbao/lsst.par` and `src/gals/swab.h` for more details.

`n_BAO` Number of BAO bins.

`n_WL` Number of WL bins.

`n_phz` Number of photo-z parameters in each component, i.e., `n_phz` photo-z bias parameters and `n_phz` photo-z rms parameters.

`sig_z0` σ_{z0}, photo-z rms error $\sigma_z = \sigma_{z0}(1+z)$.

`dndz_z0` z_0 in $n(z)$ (default 0.5).

`N_gal` Number of galaxies per square arcmin (default 50).

`cal_fsh` Calculate the BAO/WL Fisher matrix if `cal_fsh` is *true* (default *true*).

`f_name` File name of the output Fisher matrix.

Remarks The photo-z error distribution is simply a Gaussian with rms $\sigma_z = \sigma_{z0}(1+z)$ and bias $\delta z = 0$. To allow more freedom in the form of the distribution, you can modify the code to set the parameters $\sigma_{z,i}$ and δz_i individually.

The number of galaxy clustering bias parameters `nGBPar` (`src/gals/fisher.h`) is hard-wired to be the same as the number of photo-z rms parameters for convenience. You can add another command-line argument to set `nGBPar` separately.

The order of the parameters in the BAO/WL Fisher matrix is: cosmological parameters (same as CMB), galaxy clustering bias parameters (if relevant), photo-z rms parameters, and photo-z bias parameters. $N_{\rm sys}^{\rm g}$ (`DUST_NOISE`) and $N_{\rm sys}^{\gamma}$ (`SHEAR_NOISE`) are fixed in `wlbao`; `clcon` below includes them as nuisance parameters. For more generic systematics, see, e.g., Huterer et al. (2006).

Binary $P_{ij}(\ell)$ and derivatives are output from $\ell = 2$ to `SWB_L_MAX` regardless other limits on ℓ. No noise is added to $P_{ij}(\ell)$, while the shot noise is included in the derivatives with respect to photo-z parameters.

Check `OUT_DIR/CLFN/bins.txt` for information about the bins:

```
  0 S 0.001 0.701  2000 8.78086
  ...
  5 G 0.150 0.298    66 1.04568
  ...
 14 G 2.919 3.500  3000 2.07842
```

The columns are the bin ID, type (`S` for WL, and `G` for BAO), photo-$z_{i,\rm beg}$, photo-$z_{i,\rm end}$, $\ell_{i,\rm max}$, and $\bar{n}_i$ (arcmin^{-2}). Note that WL bins start from $z_{\rm p}$ = `Z_MIN` while BAO bins from `BAO_Z_MIN` (`src/gals/swab.h`).

Program `bin/clcon`

Purpose To calculate constraints on cosmological and other parameters from pre-calculated galaxy and shear power spectra.

Syntax `clcon parFile n_bin n_phz` [`sig_dz sig_gb BAO_sys WL_sys use_BAO use_WL use_tom sig_z0 l_min WL_l_ma no_nois cmb_fsh z_max f_name`]

`parFile` Name of the BAO/WL parameter file, same as that of `wlbao`.

`n_bin` Number of BAO and WL bins.

`n_phz` Number of photo-z parameters in each component, same as that of `wlbao`.

`sig_dz` Prior on photo-z bias parameters (default 0.05), $\sigma_{\rm P}(\delta z_i) =$ `sig_dz` $\times \sigma_{z0}(1+z_i)$. For convenience, $\sigma_{\rm P}(\sigma_{z,i}) = \sqrt{2}\sigma_{\rm P}(\delta z_i)$.

`sig_gb` Prior on galaxy bias parameters (default 0.15), $\sigma_{\rm P}(b_i) =$ `sig_gb` $\times\, b_i$.

`BAO_sys` `DUST_NOISE` (default set in `parFile`).

`WL_sys` `SHEAR_NOISE` (default set in `parFile`).

`use_BAO` Use galaxy auto power spectra if `use_BAO` is *true* (default *true*).

`use_WL` Use shear power spectra if `use_WL` is *true* (default *true*). Include galaxy-shear power spectra as well if both `use_BAO` and `use_WL` are true.

`use_tom` Use galaxy cross power spectra if `use_tom` is *true* (default *true*).

`sig_z0` σ_{z0} (default 0.05).

`l_min` `CTOM_L_MIN` (default `SWB_L_MIN`), Minimum ℓ for both BAO and WL, see `src/gals/ctom.h` & `ctom.cpp`.

`WL_l_ma` `WL_L_MAX` (default 2000). For convenience, I set `SWB_L_MAX` = `WL_L_MAX`+1000 in `src/clcon.cpp`.

`no_nois` No shot noise or systematics if `no_nois` is *true* (default *false*).

`cmb_fsh` File name of the CMB Fisher matrix (default `Planck.fsh`).

`z_max` `Z_MAX` (default set in `parFile`).

`f_name` File name of the output Fisher matrix if present.

Remarks Command-line arguments override those in the BAO/WL parameter file, `parFile`. However, the values of `WL_L_MAX`, `SWB_L_MAX`, and `Z_MAX` will be adjusted not to exceed those in `parFile`. `n_bin`, `n_phz`, and `sig_z0` should match the arguments supplied to `wlbao`.

Pre-calculated galaxy and shear power spectra must have the prefix `OUT_DIR/CLFN`, and `cmb_fsh` should be a properly formated CMB Fisher matrix such as generated by `cmbfm`.

You can set the priors on the photo-z parameters individually without assuming the (1+z) scaling. Same for the galaxy bias parameters. If `no_nois` is *true*, the photo-z parameters should be fixed.

The parameter order in the Fisher matrix is the same as that from `wlbao` with additions of $N_{\rm sys}^{\rm g}$ (`DUST_NOISE`) and $N_{\rm sys}^{\gamma}$ (`SHEAR_NOISE`) when relevant.

5 Hacking CSWAB

I discuss several useful components of CSWAB in this section for those who wish to customerize these components for other applications.

5.1 Global Variables

The global variables are resulted mostly from quick fixes to the ever-rising demand for new functionalities. Though convenient, they can be a hassle when one tries to modify the code. You have encountered many of them above. See `src/cmb/cmb.h`, `src/gals/swab.h`, `src/gals/angps.h`, and `src/gals/phzbin.h` for complete declarations and some brief explanations of the globals.

5.2 `class CosParEvol`

This class is declared in `src/cos/cospar.h`. It is used to calculate cosmological parameters and other quantities, such as the distance D and growth rate G, as functions of redshift. To (re)set the cosmological model for a `CosParEvol` object, call the member function

```
void Set( double Om, double Ox, double Ob, double eos0, double eosa,
          char *wFile, double h, double s8, double aMi, int tabLen );
```

where the arguments are $\Omega_{\rm M}$, $\Omega_{\rm X}$, $\Omega_{\rm B}$, w_0, w_a, the input file name of the dark energy EOS, h, σ_8, the minimum expansion factor of the interpolation table for G, D, & H, and the length of the interpolation table. If `wFile != NULL`, w_0 and w_a are ignored, and the dark energy EOS is quadratically interpolated from the file `wFile`. This file must contain two columns, the expansion factor ($0 \le a \le 1$) and the EOS. Entries do not have to be ordered, and there is no need to specify

the number of entries. Separately, if $0 <$ `aMi` ≤ 1 *and* `tabLen` ≥ 3, all subsequent results of G, D, & H will be quadratically interpolated in the range `aMi` $\leq a \leq 1$ until another `Set(...)` reverts the behavior. The following code fragment gives an example:

```
CPE  cpe;
cpe.Set(0.3, 0.7, 0.04, -1., 0., NULL, 0.73, 0.85, 0., 0);
distance = cpe.AngularDist_z(0.2, 1.0);  // D_A(z1, z2)
growth   = cpe.Growth_z(0.8);
hub_par  = cpe.Hubble_z(1.4);
curv_par = cpe.OmegaK_z(0.3);
```

where `CPE` is defined by `typedef CosParEvol CPE`. Note that the distance is always in units of h^{-1}Mpc and that the growth rate is normalized to $G(a = 1) = 1$ regardless the cosmological model. If you want to follow the convention that gives $G(a) = a$ for an Einstein-de Sitter universe, use `cpe.GetG0() * cpe.Growth_z(z)`.

The copy constructor and assignment operator have been implemented, so that you can make deep copies of a `CosParEvol` object, e.g., `cpe1 = cpe`, without worrying about the dynamically allocated elements. However, care must be taken to communicate such an object through MPI (I did not implement it). You can output the object with `cout << cpe << endl;`.

5.3 class CMBFisherMatrix

There is a useful global function worth mentioning in `src/cmb/cmb.h`:

```
void DCOS2CMBF( double cmbf[], COS_PAR i, DPAR_DIR dqDir );
```

As the name suggests, it translates the set of cosmological parameters to CMBFAST input parameters in the array `cmbf`. If `dqDir == DPAR_FID` or `i >= NUM_COS_PAR`, then `cmbf` contains the fiducial CMBFAST parameters. Otherwise, the `i`-th cosmological parameter will be perturbed in the direction `dqDir` (`DPAR_MINUS` or `DPAR_PLUS`) by the amount `DCOS[i]` before the whole set is converted to the CMBFAST parameters. The order of the CMBFAST parameters is w_0, w_a, $\Omega_{\rm C}$, $\Omega_{\rm B}$, $\Omega_{\rm X}$, Ω_ν, H_0, $Y_{\rm p}$, τ, $n_{\rm s}$, α, tensor-scalar ratio, and Δ^2_R. Note that nothing is done to Ω_ν and the tensor-scalar ratio.

The following two member functions of `CMBFisherMatrix` perform the essential tasks of initializing class members and calculating the Fisher matrix:

```
void Init( char prefix[], double f_sky );
void FisherMatrix( bool cal_Cls = false, char cmbfast[] = CMBF_EXEC );
```

where `prefix` is the prefix, e.g., the directory name, of the input/output CMB power spectra and transfer functions. The other arguments have the same meaning as those in the command-line arguments of `cmbfm`. The constraints can be checked with `void Constraints( double cons[] = NULL )`, and the Fisher matrix can be written to file with `void WriteFM( char fName[] )`.

5.4 class BAOWL

This class is declared in `src/gals/fisher.h`. It was originally designed to compute the galaxy/shear power spectra and forecast constraints at the same time, which is evident from the many Fisher matrix related member functions. The class `CosmicTomography` later takes over the forecasts with greater flexibility (see below), so that `BAOWL` is now primarily used to output the power spectra and derivatives.

To start the calculations correctly, you need to call a series of member functions in the following order to initialize the members properly:

```
void InitCosmoParams();
void InitDnDz( double N_gal, double dndz_z0 );
void InitPowerSpectrum( char prefix[] );
void InitGalaxyBias( unsigned i, double b0 = 1., double b1 = 0.84);
void InitPhotozParams( unsigned i, double sz, double dz );
void InitPhotozBins( unsigned n_BAO, unsigned n_WL );
void InitBAOWL( double f_sky );
```

where `prefix` is the prefix of transfer function names, i.e., `TFFN` in `parFile`, and `f_sky` is the effective sky fraction covered by the survey. The arguments of `InitDnDz` and `InitPhotozBins` have the same meaning as those command-line arguments of `wlbao`. The arguments of `InitGalaxyBias` are the number of parameters, $b(0)$, and $\mathrm{d}b/\mathrm{d}z$. The arguments of `InitPhotozParams` are the number of parameters in each component, σ_{z0}, and δz_0, where the photo-z rms and bias are assumed to be $\sigma_z = \mathtt{sz} \times (1+z)$ and $\delta z = \mathtt{dz} \times (1+z)$, respectively.

The galaxy/shear power spectra are calculated and written with

```
void WriteAllCl( char dir[], char prefix[] = "" );
```

where `dir` is the output directory (`OUT_DIR` in `parFile`), and `prefix` is the prefix of the file names (`CLFN` in `parFile`).

5.5 class CosmicTomography

This class is declared in `src/gals/ctom.h`. It reads the galaxy/shear power spectra and calculates the constraints. The class members are initialized with

```
void Init( char parFile[], unsigned n_bin, unsigned n_phz, unsigned l_max,
           unsigned WL_l_ma, bool use_BAO = true, bool use_WL = true,
           double z_max = 1e10 );
```

where `l_max` is assigned to `SWB_L_MAX` and other arguments have the same meaning as the command-line arguments of `clcon`. You can specify independent priors, such as the CMB Fisher matrix, with

```
void SetCMB( char cmb_fsh[], int order[], unsigned dim );
```

where the Fisher matrix in the file `cmb_fsh` has `dim` columns and `dim` rows, and `order` has `dim` elements. The `i`-th parameter in `cmb_fsh` will be associated with the `order[i]`-th parameter in my program, if $\mathtt{0} \le \mathtt{order[i]} < \mathtt{NUM\_COS\_PAR}$. Finally, the covariance matrix `cov` and Fisher matrix `fsh` are given by

```
void CovFish( double cov[], double fsh[], double ss, double sd, double sig_gb,
              double sNG = PRIOR_FLOAT, double sNS = PRIOR_FLOAT,
              bool use_tom = true ) const ;
```

The dimensions of `cov` and `fsh` are given by `unsigned NumParams()`. The array `cov` includes all the priors you have supplied, whereas `fsh` is only for BAO/WL. The prior on the photo-z rms and bias parameters are $\mathtt{ss} \times (1+z)$ and $\mathtt{sd} \times (1+z)$. `sig_gb` and `use_tom` have the same meaning as the corresponding command-line arguments. `sNG` and `sNS` are the priors on $N^g_{\rm sys}$ and $N^\gamma_{\rm sys}$, respectively. Since the data can place very strong constraints on $N^g_{\rm sys}$ and $N^\gamma_{\rm sys}$, there is no need to specify `sNG` and `sNS` unless you want to fix them.

References

Cooray, A., & Sheth, R. 2002, Phys. Rep., 372, 1

Eisenstein, D. J., & Hu, W. 1999, ApJ, 511, 5

Hu, W., & Jain, B. 2004, Phys. Rev. D, 70, 043009

Huterer, D., Takada, M., Bernstein, G., & Jain, B. 2006, MNRAS, 366, 101

Jain, B., & Bertschinger, E. 1994, ApJ, 431, 495

Jeong, D., & Komatsu, E. 2006, submitted to ApJ (astro-ph/0604075)

Knox, L., Song, Y.-S., & Zhan, H. 2006, ApJ in press (astro-ph/0605536)

Ma, Z., Hu, W., & Huterer, D. 2006, ApJ, 636, 21

Peacock, J. A., & Dodds, S. J. 1996, MNRAS, 280, L19

Schneider, M., Knox, L., Zhan, H., & Connolly, A. 2006, submitted to ApJ (astro-ph/0606098)

Seo, H.-J., & Eisenstein, D. J. 2005, ApJ, 633, 575

Song, Y.-S., & Knox, L. 2004, Phys. Rev. D, 70, 063510

Spergel, D. N. et al. 2006, submitted to ApJ (astro-ph/0603449)

Springel, V. et al. 2005, Nature, 435, 629

White, M. 2005, Astropart. Phys., 24, 334

Zaldarriaga, M., & Seljak, U. 2000, ApJS, 129, 431

Zhan, H. 2006, J. Cosmol. Astropart. Phys., JCAP08(2006)008 (astro-ph/0605696)