nicaea Documentation

Release

Martin Kilbinger

Apr 25, 2018

Contents

Contents:

CHAPTER 1

nicaea

NumerIcal Cosmology And lEnsing cAlculations

Version 2.7.2 (04/2018)

Web page: https://github.com/CosmoStat/nicaea

1.1 Authors:

Martin Kilbinger

Karim Benabed

Jean Coupon (HOD, halomodel)

Henry J. McCracken (HOD)

Liping Fu (decomp_eb)

Catherine Heymans (intrinsic alignment)

Francois Lanusse (enhancements and interface)

1.1.1 Documentation

nicaea is the cosmology part of cosmo_pmc, which can be downloaded for free at http://www.cosmopmc.info. From that site, the cosmo_pmc manual is available for further information about nicaea, which are more detailed than covered in this readme.

1.1.2 References

To reference nicaea, please use the publication *[13]* (https://arxiv.org/abs/0810.5129), in which something that resembles the first version of nicaea has been used.

1.1.3 Download, compile, and run nicaea

1.1.3.1 Download the code

Reommended: Clone the most recent stable version from github with:

```
git clone https://github.com/CosmoStat/nicaea
```

Alternatively, download the file `nicaea_<version>.tgz` from http://cosmostat.org/nicaea and un-tar the archive.

The packages fftw3 and gsl are required to compile and run nicaea. You can install fftw3 from http://www.fftw.org, and gsl from www.gnu.org/software/gsl.

1.1.3.2 Compile and install the code

Two options to compile nicaea exist. If nicaea is to be used as a library, option 1 is recommended.

Option 1: using cmake, *recommended*:

```
cd build
cmake ..
make && make install
```

The last command will copy the executable demo programs (e.g. lensingdemo) to `<BASE>/bin`, the library `libnicaea.a` to `<BASE>/lib`, and the include files to `<BASE>/include/nicaea`. The default base directory is `<BASE>=nicaea_<version>`.

If the necessary libraries are found on the system, the python module pynicaea is also installed.

The code can be tested with:

```
ctest -vv
```

To run the demo programs (see below), go to `nicaea_<version>/par_files`.

Option 2: using make.:

```
cd Demo
make
```

If fftw3 and gsl are not installed in a standard directory (e.g. `/usr`, `/usr/local`), set the variables 'FFTW' and 'GSL' in the Makefile. The header file `fftw3.h` is looked for in `$(FFTW)/include` and `libfftw3.a` in `$(FFTW)/lib`. The gsl header files are looked for in `$(GSL)/include`, the libraries `libgsl.a` and `libgslcblas.a` in `$(GSL)/lib`.

Various demo programs can be run in `./Demo`, see below.

1.1.3.3 Run the demo programs

The demo programs need parameter files in the working directory, which can be found in `par_files`.

Program name	Category	Functionality
lensingdemo	Weak lensing	density- and lensing power spectrum, lensing second-order functions
sn1demo	SNIa	Luminosity distance, distance module
halomodeldemo	Halo model	Power spectrum
cmb_bao_demo	CMB, BAO	geometrical quantities, e.g. sound horizon, angular diameter distance
decomp_eb_demo	Weak lensing	E-/B-mode decomposition (generaalized ring statistic)
cosebi_demo	Weak lensing	E-/B-mode decomposition (COSEBIs)
third_order_demo	Weak lensing	Third-order aperture-mass moments

1.1.4 Main functions

The main functions listed below have as some of their parameters:

Name	Type	Description
model	cosmo_lens* structure (see Sect. 4)	Lensing and cosmological paramaters and pre-computed tables
theta, THETA_MIN, THETA_MAX, Psimin, Psimax	double	Angular scale [rad]
R	double[3]	Array of angular scale tripes
i_bin, j_bin, k_bin	int	Redshift bin indices
err	error* (see Sect. 4)	Error structure
n	integer	COSEBIs mode.
path	string	Path to COSEBIs files with zeros for given Psimin and Psimax. Default is */path/to/nicaea/par_files/COSEBIs/*.
B_cosebi	double*	On output, B_mode is written to this pointer if non zero.
aa	array of doubles	Pre-calculated array of coefficients, see decomp_eb.c.
N	integer	Polynomial order, default 6
poly	poly_t enumeration type	Polyonmial type, default *cheby2*
wfilter	filter_t enumeration	Aperture-mass filter type, see lensing_3rd.h, default *fgauss*.
a	double	Scale factor, max(0.01,1/(1+zmax))<=a<1.0
k	double	3d Fourier wave-mode in h/Mpc
s	double	2d Fourier wave-mode, 1e-2<=ell<=1e6
ell	integer	2D harmonic mode, ell>=2

The value of the corresponding two- and three-point function is returned as double.

1.1.4.1 Second-order shear statistics

The following functions are not defined if sprojection==full.

Two-point correlation function xi+ (pm=0) and xi- (pm=1) at angular scale theta [rad]:

```
xi(model, pm, theta, i_bin, j_bin, err)
```

Top-hat shear variance in a circle of radius theta [rad]:

```
gamma2(model, theta, i_bin, j_bin, err)
```

Aperture-mass variance, polynomial filter:

```
map2_poly(model, theta, i_bin, j_bin, err)
```

Aperture-mass variance, Gaussian filter:

```
map2_gauss(model, theta, i_bin, j_bin, err)
```

COSEBIs (Complete Orthogonal E-/B-mode Integrals), *[17]*:

```
E_cosebi(model, n, Psimin, Psimax, i_bin, j_bin, path, B_cosebi, err)
```

'Ring' statistics, with Chebyshev-filter function decomposition, see *[7]*:

```
RR(model, THETA_MIN, THETA_MAX, aa, N, poly, pm, err)
```

1.1.4.2 Third-order shear statistics

Third-order aperture-mass generalized moment, *[5]*:

```
map3(model, R, i_bin, i_bin, k_bin, wfilter, err)
```

1.1.4.3 Power spectra

3d power spectrum of delta:

```
P_NL(model, a, k, err)
```

2d shear power spectrum: Pshear or Pshear+Pg^(1) if reduced-shear correction is switched on with key "sreduced = K10" in cosmo_lens.par parameter file. Returns error if sprojection==full:

```
Pshear(model, s, i_bin, j_bin, err)
```

2d shear power spectrum Pshear for integer ell. Computes full spherical projection for sprojection==full (Kilbinger et al. 2017). Calls Pshear for other cases of sprojection:

```
Pshear_spherical(model, ell, i_bin, j_bin, err)
```

2d reduced-shear correction power spectrum Pg^(1), see Kilbinger (2010). The totel (reduced-shear) power spectrum is Pkappa + Pg1:

```
Pg1(model, s, i_bin, j_bin, err)
```

1.1.4.4 Ranges

The range for k is unlimited except for the coyote10 and coyote13 non-linear emulators. For k<3.3e-6 h/Mpc and k>333 h/Mpc, the power spectrum is extrapolated (see below). The limits can be changed in cosmo.h.

The reduced-shear correction fits are accurate to 2% beetween ell=0.1 and 2*10^5. Outside that range, Pg^(1) return zero.

The range for theta is very, very large, it is determined in the routine xi_via_hankel. Although the Hankel transform is accurate only on a much smaller interval, the range of acceptable results is still from sub-arcseconds to a couple of degrees.

The limited range of the reduced-shear correction reflects in a smaller valid angular range of xi+ and xi-. If the reduced-shear is switched on, the ranges within which the second-order functions are affected to small fractions of a percent are:

Function	Minimum scale	Maximum scale
xi+	0.1'	1000'
xi-	0.5'	1000'
mapsqr	0.2'	1000'
gammasqr	0.1'	1000'
mapsqr_gauss	0.1'	1000'

1.1.5 Cosmology

The cosmology is encoded in the structure cosmo. It contains all relevant cosmological and nuisance parameters, and pre-calculated tables and constants. If parameters change, these tables are recomputed once they are needed. All lensing-related variables are contained in the structure cosmo_lens.

1.1.5.1 Reading parameters from a file

The function:

```
read_cosmological_parameters_lens(&model, F, err)
```

reads cosmological and lensing parameters from the file F (type FILE*) and initialised the structure cosmo_lens *model. The file 'cosmo_lens.par' is an example file. First, it contains a reference to the basic cosmology file 'cosmo.par', containing cosmological parameters. Next, redshift information is read from the file 'nofz.par'. Then, the lensing parameters follow.

1.1.5.2 Initializing the cosmology

The function:

```
init_parameters_lens(...)
```

returns a pointer to the structure cosmo_lens with parameters given by the arguments and blank tables. If passed to a function (e.g. one described in Sect.2), the corresponding tables and constants (if required) are filled and calculated. Successive calls to this function will be very fast since only a linear interpolation of the tabulated values is performed.

1.1.5.3 Changing the cosmology

If a different cosmology is required, a new cosmo_lens pointer has to be created, either with:

```
model_new = init_parameters_lens(...)
```

as above, or with:

```
model_new          = copy_parameters_lens_only(model, err).
model_new->param1 = ...
model_new->param2 = ...
...
```

In both cases, all tables and constants are blanked. A call of:

```
updateFrom_lens(model_new, model, err)
```

copies tables from model to model_new if corresponding parameters are unchanged and leaves those blank which have to be recalculated if required. This is particularly efficient if only a few or only "fast" parameters change since a small number of (time-consuming) functions will be recalculated. E.g., if only the redshift parameters change, the non-linear power spectra and growth factor need not be recalculated, only the shear statistics, which is very fast due to the Hankel transform.

1.1.5.4 Parameters and ranges

The following parameters are implemented. Within a given range, the program should obtain reasonable results or return an error message (see Sect.4). The program does not check whether a parameter is within its range. The following ranges have been tested some time ago, probably the code will work outside of these ranges as well.

1.1.5.5 Cosmology

Parameter	Description	Minimum	Maximum
Omega_m	total matter density (baryonic + dark)	0.1	1.5
Omega_de	dark energy density	0.1	1.5
w0_de	dark energy eos parametrization (see below)	-2.0	-0.5
w1_de	dark energy eos parametrization (see below)	-0.6	0.6
h_100	Hubble parameter H_0 = 100 h_100 km/s/Mpc	0.4	1.0
Omega_b	baryon density	0.02	0.06
Omega_nu_mass	massive neutrino density	(not tested)	
N_eff_mass	Number of massive neutrinos	(not tested)	
sigma_8	Late-time power spectrum normalisation	0.1	1.5
A_s	CMB power spectrum normalization	(not tested)	
n_spec	primordial spectral index	0.7	1.3

The power spectrum normalisation can be chosen with the flag normmode = 0 for sigma_8 and 1 for A_s.

1.1.5.6 Redshift parameters

The number of redshift bins is Nzbin. For each bin n_bin, the number of redshift parameters is given by Nnz[n_bin], its base type by nofz[n_bin]. The photometric redshift error type is photz[n_bin]. The sub-array par_nz[n_bin*Nn_max .. n_bin*Nnz_max+Nnz[n_bin]] contains the Nnz[n_bin] redshift parameters of bin n_bin. For all types the first two parameters define the minimum and maximum redshift: par_nz[n_bin*Nn_max] = zmin par_nz[n_bin*Nn_max+1] = zmax. The number of parameters is the sum of base type *Nnz_base* and photometric redshift error type parameters *Nnz_photz*.

The number of galaxies at redshift z from bin i is given by

$$n_i(z) \propto \int_{z_{\mathrm{p},i}}^{z_{\mathrm{p},i+1}} \mathrm{d}z p(z, z_\mathrm{p}) n(z)$$

and the distribution for each bin is normalized to unity.

The following base types exist:

nofz	Nnz_base	parameters	symbols	n(z) (for zmin<z<zmax)
ludo	5	alpha_p, beta_p, z0	α_p, β_p, z_0	$(z/z_0)^{\alpha_p} \exp[-(z/z_0)^{\beta_p}]$
jonben	5	a, b, c	a, b, c	$z^a/(z^b + c)$
ymmk	5	a, b, c	a, b, c	$(z^a + z^{ab})/(z^b + c)$
cfhtlens	7	z1, z2, ac, b, d	$z_1, z_2, a/c, b, d$	$a/c * \exp(-((z - z_1)/b)^2) + \exp(-((z - z_2)/d)^2)$
single	2	z0	z_0	$\delta_{\rm D}(z - z0)$
hist	2n+1	zi, Ni	$z_0 \ldots z_n, N_0 \ldots N_{n-1}$	Histogram with n bins of values N_i and corners z_i

type=hist assumes a N(z) histogram with n bins.

The following photometric redshift error types exist:

photz	Nnz_photz	parameters	symbols	p(z, z_p)
photz_no	0	•	•	$\delta_{\rm D}(z - z_{\rm p})$
photz_gauss	7	sigma_z, z_bias, c_cal, f_out, sigma_z_out, z_bias_out, c_cal_out	$\sigma_z, z_{\rm b}, c_{\rm cal}, f_{\rm out}, \sigma_{z,\rm out}, z_{\rm b,out}, c_{\rm cal,out}$	$(1 - f_{\rm out})\mathcal{N}(z, c_{\rm cal} z_{\rm p} - z_{\rm b}, \sigma_z(1 + z_{\rm p})) + f_{\rm out}\mathcal{N}(z c_{\rm cal,out} z_{\rm p} - z_{\rm b,out}, \sigma_{z,\rm out}(1 + z_{\rm p}))$

$\mathcal{N}(\mu, \sigma)$ is a Gaussian with mean μ and variance σ.

The parameters are stored in the vector par_nz as follows:

0	1	2	3	...	n	n+1	n+2	...	2n
z_0	z_n	z_1	z_2	...	z_{n-1}	N_0	N_1	...	N_{n-1}

The number of parameters is $N_{n_z} = 2n + 1$. The redshifts z_i are understood as the lower bin boundaries with the exception of $z_n = z_{\rm max}$ which is the limiting redshift. The i-th bin therefore is between z_i and z_{i+1}, the (unnormalized) number of galaxies is N_i. $z_{\rm min} = z_0$ and $z_{\rm max} = z_n$ are in the first two entries, as required.

A general nofz file (except hist and single, see below) has a one-line header with the base type nofz, and optional the photometric redshift error type photz. This is followed by Nnz_base lines with the nofz parameter values, one in each line, followed by the Nnz_photz parameters if any:

nofz [photz]
p_0
p_1
...
[p_{Nnz_base-1}
q_0
...
q_{Nnz_photz-1}]

For the nofz types *hist* and *single*, photometric redshift errors cannot be defined.

nofz=hist

The function read_par_nz_hist reads the histogram data from a file, sets Nnz and returns par_nz. The file has to have the following structure:

# hist	
z_0	N_0
z_1	N_1
...	...
z_{n-1}	N_{n-1}
z_n	0.0

The last redshift value z_n is the right corner of the highest redshift bin, and the corresponding number of galaxies if necessarily 0.

nofz=single

All galaxies are at a single redshift z0 can achieved with the following file:

single
z0
z0

(The value z0 has to appear twice. It is both zmin and zmax.)

The normalization for all types, int_zmin^zmax prob(z) dz = 1, is calculated in the code.

1.1.5.7 Flags

key	value	reference
nonlinear	linear	Linear power spectrum (*[1]* CDM transfer function)
	pd96	*[4]* fitting formula
	smith03	Smith et al. (2003) halofit, *[18]*
	smith03_de	Smith et al. (2003) halofit + dark-energy correction from icosmo.org
	smith03_revised	Takahashi et al. (2012), revised halofit parameters, [2012ApJ...761..152T]
	coyote10	Coyote emulator v1, *[10]*, *[8]*, *[15]*
	coyote13	Coyote emulator v2, *[9]*
transfer	bbks	Bardeen et al. (1986) transfer function, *[1]*
	eisenhu	Eisenstein & Hu (1998) "shape fit", [1998ApJ...496..605E]
	camb	Using camb for T(k) (not yet supported)
growth	heath	Heath (1977) analytical expression for linear growth factor (valid only for no or a pure cosmologi
	growth_de	General dark energy model
de_param	jassal	$w(a) = w_{0,\mathrm{de}} + w_{1,\mathrm{de}} a(1-a)$
	linder	$w(a) = w_{0,\mathrm{de}} + w_{1,\mathrm{de}}(1-a)$
	earlyDE	$w(a) = w_{0,\mathrm{de}} / \sqrt{1 - b_{\mathrm{early}} \log a}$
normmode	0	normalization = σ_8

Table 1.1 – continued from previous page

key	value	reference
	1	normalization = A_S
tomo	tomo_all	All redshift-correlations (ij), i<=j
	tomo_auto_only	Only autos-correlations (ii)
	tomo_cross_only	Only cross-correlations (i!=j)
sprojection	limber	Standard 1st-order flat-sky Limber approximation, L1Fl
	limber_la08	*Depreciated*: Extended 1st-order flat-sky Limber, ExtL1Fl, *[16]*
	limber_la08_hyb	Extended 1st-order flat-sky hybrid Limber, ExtL1FlHyb
	limber_la08_sph	Extended 1st-order spherical Limber, best 1st-order approx., ExtL1Sph
	limber2_la08	*Depreciated*: Extended 2nd-order flat-sky Limber, ExtL2Fl
	limber2_la08_hyb	Extended 2nd-order flat-sky Limber hybrid, ExtL2FlHyb
	limber2_la08_sph	Extended 2nd-order spherical Limber, best approx., ExtL2Sph, *[14]*
	full	Full spherical projection, slow, not for real-space functions
reduced	none	No reduced-shear correction
	K10	Reduced-shear according to *[12]*
q_mag_size	double	If reduced==K10: q_mag_size = 2*(alpha+beta-1), see K10 eq. 16. Set q_mag_size = 0 if no mag
sia	none	No intrinsic alignment (IA)
	HS04	Hirata & Seljak linear IA model, *[11]*
sia_terns	none	No IA
	GI_II	If sia!=none: add GI and II (standard IA)
	only_GI	If sia!=none: only add GI
	only_II	If sia!=none: only add II
A_ia	double	If sia!=none: Global amplitude of IA contribution.

The range for w0_de and w1_de correspond to de_param=linder.

The minimum scale factor a_min (used for various integrations) is set using the function set_amin().

1.1.6 Errors and diagnostics

Most of the situations where an error or undefined value occurs are intercepted by the program. In that case, a variable *err of type error* is set via the macros:

```
*err = addError(error_type, "message", *err, __LINE__)
```

or:

```
*err = addErrorVA(error_type, "formatted message", *err, __LINE__, VA_LIST)
```

storing the line in the code, a message and the error type (ce_xyz). With:

```
testErrorRet(test, error_type, "message", *err, __LINE__, return_value)
```

or:

```
testErrorRetVA(test, error_type, "formatted message", *err, __LINE__, return_value,␣
↪VA_LIST)
```

a conditional error is produced if the (Boolean) expression test is true. The error can be transported up the stack to the calling function with the macro:

```
forwardError(*err, __LINE__, return_value)
```

(in case of a void function omit *return_value* but keep the comma before the closing bracket). This can be used as diagnostics even for errors deep in the hierarchy of functions. To exit on an error, use:

```
exitOnError(*err, FILE)
```

At the start of the program, or after an error had occurred but one wishes to continue, maybe with a different cosmology, set:

```
*err = NULL
```

An error can be caused by undefined values, not initialized parameters, function arguments outside the valid range. Further, a specific cosmology may not allow certain functions to be carried out. For example, in a loitering Universe there is a maximum redshift, and if the redshift distribution extends over this maximum, the angular diameter distance is undefined and an error is produced.

1.1.7 Extrapolation

In the highly non-linear regime, the power spectrum is extrapolated. For the linear power spectrum, $P(k) \propto k^{n_\mathrm{s}-4.0}$ is assumed. In the PD96-case, the stable clustering result $P(k) \propto k^{-2.5}$ is used. For halofit, the asymptotic form of the halofit formula is taken, see Rob's paper eq. (61).

In the linear regime at small k, the extrapolation is $P(k) \propto k_\mathrm{s}^n$.

1.1.8 Performance

Time-consuming functions store tabulated values and interpolated when called after the first time. The tables are recalculated when cosmological parameters have changed since the previous call. The correlation functions are calculated using a fast Hankel transform.

1.1.9 Known bugs and shortcomings

- Some parameter combinations cause undefined behaviour of the program. These are (hopefully) intercepted and an error is created (see Sect. 5). E.g., for n_spec<0.7, f_NL *[4]*) is not defined. For a closed Universe, the probed redshift can be larger than the maximum redshift.
- a=1.0 very rarely creates an error, use 0.99999... instead.
- The code is not well suited for Fisher matrix calculations. In particular for the inverse Fisher matrix, numerical derivatives have to be very accurate, and the interpolations between tabulated values (linear and spline) in nicaea introduce numerical noise that can render the Fisher matrix numerically singular :cite'WKWG12'.
- Dark-energy models, in particular with varying w(z), are not recommended for the non_linear models smith03, and smith03_de. Instead, use the revised halofit model with smith03_revised.

In case of problems please don't hesitate to contact me at martin.kilbinger@cea.fr . Questions and comments are welcome!

1.1.10 Changes compared to the Rob Smith's original halofit

Parts of the program 'cosmo.c' is based on Rob Smiths' halofit *[18]*. The code for determining the non-linear power spectrum has been improved and made more efficient. The main changes are listed below. The code also includes the non-linear fitting formulae of *[4]*.

- Tabulation of the linear and non-linear power spectrum, constants are calculated only once.

- Integration cutoff for determination of non-linear scale knl flexible, as function of smoothing scale rmid; using Romberg integration.
- Bisection to find knl is iterative: if the bisection gets stuck at one end of the bisecting interval, the interval is shifted accordingly and a new bisection is started. If knl is larger than knlstern (I chose 10^6 h/Mpc), the bisection is canceled and the linear power spectrum is used.
- Slope and curvature are calculated only once, after knl is fixed.
- The Eisenstein & Hu (1998) [1998ApJ...496..605E] fit for the transfer function is used instead of Bond&Efstathiou (1984).
- The exact linear growth factor is used instead of the *[2]* fitting formula. Dark energy models are incorporated.

1.1.11 Acknowledgements

We thank Alexandre Boucaud, Jan Hartlap, Alina Kiessling, Jasmin Pielorz, Peter Schneider, Rob E. Smith, Patrick Simon, Masahiro Takada, Melody Wolk, and the CosmoSIS development team for helpful suggestions.

1.1.12 References

1.1.13 Contact

Feel free to email me at martin.kilbinger@cea.fr

Have fun! Martin Kilbinger

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Bibliography

[1] J. M. Bardeen, J. R. Bond, N. Kaiser, and A. S. Szalay. The statistics of peaks of gaussian random fields. *apj* , 304:15–61, 1986.

[2] S. M. Carroll, W. H. Press, and E. L. Turner. The cosmological constant. *araa* , 30:499–542, 1992.

[3] D. J. Heath. The growth of density pertubations in zero pressure friedmann-lemaître universe. *mnras* , 179:351, 1977.

[4] J. A. Peacock and S. J. Dodds. Nonlinear evolution of cosmological power spectra. *mnras* , 280:L19, 1996.

[5] P. Schneider, M. Kilbinger, and M. Lombardi. The three-point correlation function of cosmic shear. II: relation to the bispectrum of the projected mass density and generalized third-order aperture mass. *aap* , 431:9–25, 2005. arXiv:astro-ph/0308328.

[1998ApJ...496..605E] D. J. Eisenstein and W. Hu. Baryonic Features in the Matter Transfer Function. *apj* , 496:605, March 1998. ads link.

[7] L. Fu and M. Kilbinger. A new cosmic shear function: Optimised E-/B-mode decomposition on a finite interval. *mnras* , 401:1264–1274, 2010. ads link.

[8] K. Heitmann, D. Higdon, M. White, S. Habib, B. J. Williams, E. Lawrence, and C. Wagner. The Coyote Universe. II. Cosmological Models and Precision Emulation of the Nonlinear Matter Power Spectrum. *apj* , 705:156–174, November 2009. ads link, arXiv:0902.0429, doi:10.1088/0004-637X/705/1/156.

[9] K. Heitmann, E. Lawrence, J. Kwan, S. Habib, and D. Higdon. The Coyote Universe Extended: Precision Emulation of the Matter Power Spectrum. *apj* , 780:111, January 2014. ads link, arXiv:1304.7849, doi:10.1088/0004-637X/780/1/111.

[10] K. Heitmann, M. White, C. Wagner, S. Habib, and D. Higdon. The Coyote Universe. I. Precision Determination of the Nonlinear Matter Power Spectrum. *apj* , 715:104–121, 2010. ads link, arXiv:0812.1052, doi:10.1088/0004-637X/715/1/104.

[11] C. M. Hirata and U. Seljak. Intrinsic alignment-lensing interference as a contaminant of cosmic shear. *prd* , 70(6):063526–+, September 2004. ads link.

[12] M. Kilbinger. Fitting formulae of the reduced-shear power spectrum for weak lensing. *aap* , 519:A19+, 2010. ads link, arXiv:1004.3493, doi:10.1051/0004-6361/201014829.

[13] M. Kilbinger, K. Benabed, J. Guy, P. Astier, I. Tereno, L. Fu, D. Wraith, J. Coupon, Y. Mellier, C. Balland, F. R. Bouchet, T. Hamana, D. Hardin, H. J. McCracken, R. Pain, N. Regnault, M. Schultheis, and H. Yahagi.

Dark-energy constraints and correlations with systematics from CFHTLS weak lensing, SNLS supernovae Ia and WMAP5. *aap* , 497:677–688, 2009. ads link, arXiv:0810.5129, doi:10.1051/0004-6361/200811247.

[14] M. Kilbinger, C. Heymans, M. Asgari, S. Joudaki, P. Schneider, P. Simon, L. Van Waerbeke, J. Harnois-Déraps, H. Hildebrandt, F. Köhlinger, K. Kuijken, and M. Viola. Precision calculations of the cosmic shear power spectrum projection. *mnras* , 472:2126–2141, 2017. ads link, arXiv:1702.05301, doi:10.1093/mnras/stx2082.

[15] E. Lawrence, K. Heitmann, M. White, D. Higdon, C. Wagner, S. Habib, and B. Williams. The Coyote Universe. III. Simulation Suite and Precision Emulator for the Nonlinear Matter Power Spectrum. *apj* , 713:1322–1331, April 2010. ads link, arXiv:0912.4490, doi:10.1088/0004-637X/713/2/1322.

[16] M. LoVerde and N. Afshordi. Extended Limber approximation. *prd* , 78(12):123506, December 2008. ads link, arXiv:0809.5112, doi:10.1103/PhysRevD.78.123506.

[17] P. Schneider, T. Eifler, and E. Krause. COSEBIs: Extracting the full E-/B-mode information from cosmic shear correlation functions. *aap* , 520:A116, September 2010. ads link, arXiv:1002.2136, doi:10.1051/0004-6361/201014235.

[18] R. E. Smith, J. A. Peacock, A. Jenkins, S. D. M. White, C. S. Frenk, F. R. Pearce, P. A. Thomas, G. Efstathiou, and H. M. P. Couchman. Stable clustering, the halo model and non-linear cosmological power spectra. *mnras* , 341:1311–1332, June 2003. ads link.

[2012ApJ...761..152T] R. Takahashi, M. Sato, T. Nishimichi, A. Taruya, and M. Oguri. Revising the Halofit Model for the Nonlinear Matter Power Spectrum. *apj* , 761:152, December 2012. ads link, arXiv:1208.2701, doi:10.1088/0004-637X/761/2/152.