Input and output files
Basic running

Input files

./class can take two input files *.ini and *.pre:

```
>./class my_model.ini some_precision.pre
```

But one is enough. Syntax:

```
h = 0.7
T_cmb = 2.726 # comment
output = tCl, pCl
more comments, ignored because there is no equal sign
# comment with an =, still ignored thanks to the sharp
```

Input files

`./class` can take two input files `*.ini` and `*.pre`:

```
>./class my_model.ini some_precision.pre
```

But one is enough. Syntax:

```
h = 0.7
T_cmb = 2.726 # comment
output = tCl, pCl
more comments, ignored because there is no equal sign
# comment with an =, still ignored thanks to the sharp
```

- Order of lines doesn't matter at all.
- All parameters not passed fixed to default, i.e. the most reasonable or minimalistic choice
- All possible input parameters and details on the syntax explained in `explanatory.ini`
- This is only a reference file; we advise you *never* to modify it, but rather to copy it and reduce it to a shorter and more friendly file.
- For *basic* usage: `explanatory.ini` $\equiv$ full documentation of the code

Input files

For instance, we can create a very short file `lcdm.ini`:

```
******************************
* CLASS input parameter file *
******************************
----> background parameters:
H0 = 72.
omega_b = 0.0266691
omega_cdm = 0.110616
----> thermodynamics parameters:
z_reio = 10.
----> define primordial perturbation spectra:
A_s = 2.3e-9
n_s = 1.
----> define which perturbations should be computed:
output = tCl, pCl
----> parameters for the output spectra:
l_scalar_max = 2500
```

Input files

Try to run the code with an even smaller input file `nut.ini`:

```
output = tCl
output_verbose=1
```

Run with

```
./class nut.ini
```

Check that C_l's have been written in `output/nut00_cl.dat`

```
more output/nut00_cl.dat
```

Essential input parameters (1/2)

Essential input parameters controlling the output (see details in `explanatory.ini`):

```
modes = s,t
ic = ad, cdi, bi, nid, niv
lensing = yes
non linear = halofit
output = tCl, pCl, lCl, mPk, mTk, vTk, nCl, sCl

l_max_scalars=2500
l_max_tensors=500
l_max_lss = 1000
P_k_max_h/Mpc = 0.2
#P_k_max_1/Mpc =
z_pk = 0 #or 1,2,10

root = output/test_ #default: output/<ini_file>##_

headers = [yes/no]
format = [class/camb]
                                                    ...
```

Essential input parameters (2/2)

Essential input parameters controlling the output (see details in `explanatory.ini`):

```
write background = [yes/no]
write thermodynamics = [yes/no]
k_output_values = 0.01, 0.1, 0.0001
write primordial = [yes/no]

write parameters = [yes/no]
write warnings = [yes/no]

input_verbose = 1   # or 0, 2, 3,...
background_verbose = 1
thermodynamics_verbose = 1
perturbations_verbose = 1
transfer_verbose = 1
primordial_verbose = 1
spectra_verbose = 1
nonlinear_verbose = 1
lensing_verbose = 1
output_verbose = 1
```

Output files (1/3)

Following files created (or not) automatically (here we assume that `root=test_`), depending on the content of the input fields:

```
output= [tCl,pCl,lCl,nCl,sCl,mPk,dTk,vTk]
lensing= [yes, no]
modes = [s,v,t]
ic = [ad, bi, cdi, nid, niv]
# ... and more, e.g. for nCl, sCl redshift bins
```

- `test_cl.dat` total unlensed C_l's

Output files (1/3)

Following files created (or not) automatically (here we assume that `root=test_`), depending on the content of the input fields:

```
output= [tCl,pCl,lCl,nCl,sCl,mPk,dTk,vTk]
lensing= [yes, no]
modes = [s,v,t]
ic = [ad, bi, cdi, nid, niv]
# ... and more, e.g. for nCl, sCl redshift bins
```

- `test_cl.dat` total unlensed C_l's
- `test_cl_lensed.dat` total lensed C_l's

Output files (1/3)

Following files created (or not) automatically (here we assume that `root=test_`), depending on the content of the input fields:

```
output= [tCl,pCl,lCl,nCl,sCl,mPk,dTk,vTk]
lensing= [yes, no]
modes = [s,v,t]
ic = [ad, bi, cdi, nid, niv]
# ... and more, e.g. for nCl, sCl redshift bins
```

- `test_cl.dat` total unlensed C_l's
- `test_cl_lensed.dat` total lensed C_l's
- `test_cls.dat` scalar C_l's when two modes
- `test_clt.dat` tensor C_l's when two modes

Output files (1/3)

Following files created (or not) automatically (here we assume that `root=test_`), depending on the content of the input fields:

```
output= [tCl,pCl,lCl,nCl,sCl,mPk,dTk,vTk]
lensing= [yes, no]
modes = [s,v,t]
ic = [ad, bi, cdi, nid, niv]
# ... and more, e.g. for nCl, sCl redshift bins
```

- `test_cl.dat` total unlensed C_l's
- `test_cl_lensed.dat` total lensed C_l's
- `test_cls.dat` scalar C_l's when two modes
- `test_clt.dat` tensor C_l's when two modes
- `test_cl_ad.dat`, `test_cl_cdi.dat`, `test_cl_ad_cdi.dat` etc. when different i.c. requested

Output files (1/3)

Following files created (or not) automatically (here we assume that `root=test_`), depending on the content of the input fields:

```
output= [tCl,pCl,lCl,nCl,sCl,mPk,dTk,vTk]
lensing= [yes, no]
modes = [s,v,t]
ic = [ad, bi, cdi, nid, niv]
# ... and more, e.g. for nCl, sCl redshift bins
```

- `test_cl.dat` total unlensed C_l's
- `test_cl_lensed.dat` total lensed C_l's
- `test_cls.dat` scalar C_l's when two modes
- `test_clt.dat` tensor C_l's when two modes
- `test_cl_ad.dat`, `test_cl_cdi.dat`, `test_cl_ad_cdi.dat` etc. when different i.c. requested

Number of columns in these files can vary a lot depending on input parameters. Always indicated in the header.

Run with `root=output/test1_`, with either `output=tCl` or `tCl,pCl,lCl`, and either `format = camb` or `class`.
In each case, look at headers in `output/test_cl.dat`

Output files (2/3)

Following files created (or not) automatically (here we assume that `root=test_`), depending on the content of the input fields:

```
output= [tCl,pCl,lCl,nCl,sCl,mPk,dTk,vTk]
non linear= [none, halofit]
modes = [s,v,t]
ic = [ad, bi, cdi, nid, niv]
z_pk = [list of values]
```

- `test_pk.dat` matter power spectrum

Output files (2/3)

Following files created (or not) automatically (here we assume that `root=test_`), depending on the content of the input fields:

```
output= [tCl,pCl,lCl,nCl,sCl,mPk,dTk,vTk]
non linear= [none, halofit]
modes = [s,v,t]
ic = [ad, bi, cdi, nid, niv]
z_pk = [list of values]
```

- `test_pk.dat` matter power spectrum
- `test_pk_nl.dat` non-linear matter power spectrum

Output files (2/3)

Following files created (or not) automatically (here we assume that `root=test_`), depending on the content of the input fields:

```
output= [tCl,pCl,lCl,nCl,sCl,mPk,dTk,vTk]
non linear= [none, halofit]
modes = [s,v,t]
ic = [ad, bi, cdi, nid, niv]
z_pk = [list of values]
```

- `test_pk.dat` matter power spectrum
- `test_pk_nl.dat` non-linear matter power spectrum
- `test_pk_ad.dat`, `test_pk_cdi.dat`, `test_pk_ad_cdi.dat` etc. when different i.c. requested

Output files (2/3)

Following files created (or not) automatically (here we assume that `root=test_`), depending on the content of the input fields:

```
output= [tCl,pCl,lCl,nCl,sCl,mPk,dTk,vTk]
non linear= [none, halofit]
modes = [s,v,t]
ic = [ad, bi, cdi, nid, niv]
z_pk = [list of values]
```

- `test_pk.dat` matter power spectrum
- `test_pk_nl.dat` non-linear matter power spectrum
- `test_pk_ad.dat`, `test_pk_cdi.dat`, `test_pk_ad_cdi.dat` etc. when different i.c. requested
- `test_tk.dat` density and/or velocity transfer functions

Output files (2/3)

Following files created (or not) automatically (here we assume that `root=test_`), depending on the content of the input fields:

```
output= [tCl,pCl,lCl,nCl,sCl,mPk,dTk,vTk]
non linear= [none, halofit]
modes = [s,v,t]
ic = [ad, bi, cdi, nid, niv]
z_pk = [list of values]
```

- `test_pk.dat` matter power spectrum
- `test_pk_nl.dat` non-linear matter power spectrum
- `test_pk_ad.dat`, `test_pk_cdi.dat`, `test_pk_ad_cdi.dat` etc. when different i.c. requested
- `test_tk.dat` density and/or velocity transfer functions
- `test_tk_ad.dat`, `test_tk_cdi.dat`, `test_tk_ad_cdi.dat` etc. when different i.c. requested

Output files (2/3)

Following files created (or not) automatically (here we assume that `root=test_`), depending on the content of the input fields:

```
output= [tCl,pCl,lCl,nCl,sCl,mPk,dTk,vTk]
non linear= [none, halofit]
modes = [s,v,t]
ic = [ad, bi, cdi, nid, niv]
z_pk = [list of values]
```

- `test_pk.dat` matter power spectrum
- `test_pk_nl.dat` non-linear matter power spectrum
- `test_pk_ad.dat`, `test_pk_cdi.dat`, `test_pk_ad_cdi.dat` etc. when different i.c. requested
- `test_tk.dat` density and/or velocity transfer functions
- `test_tk_ad.dat`, `test_tk_cdi.dat`, `test_tk_ad_cdi.dat` etc. when different i.c. requested
- if `pk` or `tk` requested at different redshift, several files, with extra suffix `_z0`, `_z1`, etc.

Output files (2/3)

Following files created (or not) automatically (here we assume that `root=test_`), depending on the content of the input fields:

```
output= [tCl,pCl,lCl,nCl,sCl,mPk,dTk,vTk]
non linear= [none, halofit]
modes = [s,v,t]
ic = [ad, bi, cdi, nid, niv]
z_pk = [list of values]
```

- `test_pk.dat` matter power spectrum
- `test_pk_nl.dat` non-linear matter power spectrum
- `test_pk_ad.dat`, `test_pk_cdi.dat`, `test_pk_ad_cdi.dat` etc. when different i.c. requested
- `test_tk.dat` density and/or velocity transfer functions
- `test_tk_ad.dat`, `test_tk_cdi.dat`, `test_tk_ad_cdi.dat` etc. when different i.c. requested
- if `pk` or `tk` requested at different redshift, several files, with extra suffix `_z0`, `_z1`, etc.

Run with `root=output/test2_`, `output=mPk`,
and without or with the extra line `z_pk=0,0.4,0.8`.
In each case, look at output file names and headers.

Output files (3/3)

Following files created (or not) automatically (here we assume that `root=test_`), depending on the content of the input fields:

```
write background = [yes,no]
write thermodynamics = [yes,no]
write primordial = [yes,no]
k_output_values = [list of values]
modes = [s,v,t]
```

- `test_background.dat` background quantities versus time and redshift

Output files (3/3)

Following files created (or not) automatically (here we assume that `root=test_`), depending on the content of the input fields:

```
write background = [yes,no]
write thermodynamics = [yes,no]
write primordial = [yes,no]
k_output_values = [list of values]
modes = [s,v,t]
```

- `test_background.dat` background quantities versus time and redshift
- `test_thermodynamics.dat` thermodynamical quantities versus redshift

Output files (3/3)

Following files created (or not) automatically (here we assume that `root=test_`), depending on the content of the input fields:

```
write background = [yes,no]
write thermodynamics = [yes,no]
write primordial = [yes,no]
k_output_values = [list of values]
modes = [s,v,t]
```

- `test_background.dat` background quantities versus time and redshift
- `test_thermodynamics.dat` thermodynamical quantities versus redshift
- `test_primordial.dat` primordial spectra (may follow from inflation simulation)

Output files (3/3)

Following files created (or not) automatically (here we assume that `root=test_`), depending on the content of the input fields:

```
write background = [yes,no]
write thermodynamics = [yes,no]
write primordial = [yes,no]
k_output_values = [list of values]
modes = [s,v,t]
```

- `test_background.dat` background quantities versus time and redshift
- `test_thermodynamics.dat` thermodynamical quantities versus redshift
- `test_primordial.dat` primordial spectra (may follow from inflation simulation)
- `test_perturbations_k*_s[vt].dat` evolution of perturbations versus time

Output files (3/3)

Following files created (or not) automatically (here we assume that `root=test_`), depending on the content of the input fields:

```
write background = [yes,no]
write thermodynamics = [yes,no]
write primordial = [yes,no]
k_output_values = [list of values]
modes = [s,v,t]
```

- `test_background.dat` background quantities versus time and redshift
- `test_thermodynamics.dat` thermodynamical quantities versus redshift
- `test_primordial.dat` primordial spectra (may follow from inflation simulation)
- `test_perturbations_k*_s[vt].dat` evolution of perturbations versus time

Run with `root=output/test3_`, `output=tCl`, `k_output_values = 0.001,0.01`.
Look at output file names and headers.

Exercise 0

The exercise text is located in the Dropbox, in the `CLASS_exercise/` folder.

0

Use the information sent by CLASS to the standard output (= displayed in the terminal) to check the characteristics of the Planck best-fit model presented in the Planck2013 Cosmological Parameter paper.