Overall style and structure

class directory

The directory `class/` contains subdirectories:

```
include/ # header files (*.h) containing declarations
source/  # the 10 important modules of CLASS
main/    # main CLASS function: short, just calls 10 modules
test/    # other main functions for testing part of the code
tools/   # auxiliary pieces of codes (numerical methods)
output/  # output files
python/  # python wrapper of CLASS
cpp/     # C++ wrapper of CLASS
build/   # binary files created at compilation
```

plus examples of input files, `README.rst`, `Makefile`, and few other directories containing ancillary data (`bbn/`) or external code (`hyrec/`)

Look at these directories

```
class$ ls
class$ ls source/; ls main/
```

class modules

In CLASS, what is a module?

- a file `include/xxx.h` containing some declarations
- a file `source/xxx.c` containing some functions
- each module is a associated with a structure `xx`, containing all what other modules need to know, and nothing else
- some fields in this structure are filled in the `input.c` module (input parameters relevant for this module)
- all other fields are filled by a function `xxx_init(...)`
- "executing a module" $\equiv$ calling `xxx_init(...)`

In `include/background.h`: localise `struct background`
In `source/background.c`: localise `background_init()`

class modules

List of structures associated to modules:

module	structure	ab.	*	main content

class modules

List of structures associated to modules:

module	structure	ab.	*	main content
input.c	precision	pr	ppr	all precision parameters

class modules

List of structures associated to modules:

module	structure	ab.	*	main content
input.c	precision	pr	ppr	all precision parameters
background.c	background	ba	pba	background quantities as funct. of τ

class modules

List of structures associated to modules:

module	structure	ab.	*	main content
input.c	precision	pr	ppr	all precision parameters
background.c	background	ba	pba	background quantities as funct. of τ
thermodynamics.c	thermodynamics	th	pth	thermo. quantities as funct. of z

class modules

List of structures associated to modules:

module	structure	ab.	*	main content
input.c	precision	pr	ppr	all precision parameters
background.c	background	ba	pba	background quantities as funct. of τ
thermodynamics.c	thermodynamics	th	pth	thermo. quantities as funct. of z
perturbations.c	perturbs	pt	ppt	source functions $S(k,t)$

class modules

List of structures associated to modules:

module	structure	ab.	*	main content
input.c	precision	pr	ppr	all precision parameters
background.c	background	ba	pba	background quantities as funct. of τ
thermodynamics.c	thermodynamics	th	pth	thermo. quantities as funct. of z
perturbations.c	perturbs	pt	ppt	source functions $S(k,t)$
primordial.c	primordial	pm	ppm	primordial spectra $\mathcal{P}(k)$

class modules

List of structures associated to modules:

module	structure	ab.	*	main content
input.c	precision	pr	ppr	all precision parameters
background.c	background	ba	pba	background quantities as funct. of τ
thermodynamics.c	thermodynamics	th	pth	thermo. quantities as funct. of z
perturbations.c	perturbs	pt	ppt	source functions $S(k,t)$
primordial.c	primordial	pm	ppm	primordial spectra $\mathcal{P}(k)$
nonlinear.c	nonlinear	nl	pnl	non-linear corrections $\alpha_{\rm NL}(k,\tau)$

class modules

List of structures associated to modules:

module	structure	ab.	*	main content
input.c	precision	pr	ppr	all precision parameters
background.c	background	ba	pba	background quantities as funct. of τ
thermodynamics.c	thermodynamics	th	pth	thermo. quantities as funct. of z
perturbations.c	perturbs	pt	ppt	source functions $S(k,t)$
primordial.c	primordial	pm	ppm	primordial spectra $\mathcal{P}(k)$
nonlinear.c	nonlinear	nl	pnl	non-linear corrections $\alpha_{\rm NL}(k,\tau)$
transfer.c	transfers	tr	ptr	transfer functions $\Delta_l(k)$

class modules

List of structures associated to modules:

module	structure	ab.	*	main content
input.c	precision	pr	ppr	all precision parameters
background.c	background	ba	pba	background quantities as funct. of τ
thermodynamics.c	thermodynamics	th	pth	thermo. quantities as funct. of z
perturbations.c	perturbs	pt	ppt	source functions $S(k,t)$
primordial.c	primordial	pm	ppm	primordial spectra $\mathcal{P}(k)$
nonlinear.c	nonlinear	nl	pnl	non-linear corrections $\alpha_{\rm NL}(k,\tau)$
transfer.c	transfers	tr	ptr	transfer functions $\Delta_l(k)$
spectra.c	spectra	sp	psp	linear and/or non-linear $P(k,z)$, C_ℓ's

class modules

List of structures associated to modules:

module	structure	ab.	*	main content
input.c	precision	pr	ppr	all precision parameters
background.c	background	ba	pba	background quantities as funct. of τ
thermodynamics.c	thermodynamics	th	pth	thermo. quantities as funct. of z
perturbations.c	perturbs	pt	ppt	source functions $S(k,t)$
primordial.c	primordial	pm	ppm	primordial spectra $\mathcal{P}(k)$
nonlinear.c	nonlinear	nl	pnl	non-linear corrections $\alpha_{\rm NL}(k,\tau)$
transfer.c	transfers	tr	ptr	transfer functions $\Delta_l(k)$
spectra.c	spectra	sp	psp	linear and/or non-linear $P(k,z)$, C_ℓ's
lensing.c	lensing	le	ple	lensed C_ℓ's

class modules

List of structures associated to modules:

module	structure	ab.	*	main content
input.c	precision	pr	ppr	all precision parameters
background.c	background	ba	pba	background quantities as funct. of τ
thermodynamics.c	thermodynamics	th	pth	thermo. quantities as funct. of z
perturbations.c	perturbs	pt	ppt	source functions $S(k,t)$
primordial.c	primordial	pm	ppm	primordial spectra $\mathcal{P}(k)$
nonlinear.c	nonlinear	nl	pnl	non-linear corrections $\alpha_{\rm NL}(k,\tau)$
transfer.c	transfers	tr	ptr	transfer functions $\Delta_l(k)$
spectra.c	spectra	sp	psp	linear and/or non-linear $P(k,z)$, C_ℓ's
lensing.c	lensing	le	ple	lensed C_ℓ's
output.c	output	op	pop	description of output format

class modules

Each module contains:

- a function `xxx_init(...)` filling the structure `xx`
- a function `xxx_free(...)` freeing all the memory allocated to this structure
- some functions `xxx_external_1(...)`, ..., `xxx_external_n(...)` that can be called from other modules (e.g. to read correctly or interpolate the content of the structure `xx`)
- some functions `xxx_internal_1(...)`, ..., `xxx_internal_m(...)` that are called only inside the module, within `xxx_init(...)`

class modules

Each module contains:

- a function `xxx_init(...)` filling the structure `xx`
- a function `xxx_free(...)` freeing all the memory allocated to this structure
- some functions `xxx_external_1(...)`, ..., `xxx_external_n(...)` that can be called from other modules (e.g. to read correctly or interpolate the content of the structure `xx`)
- some functions `xxx_internal_1(...)`, ..., `xxx_internal_m(...)` that are called only inside the module, within `xxx_init(...)`

Following order always respected in `xxx.c`:

```
xxx_external_1(...)
...
xxx_external_n(...)
xxx_init(...)
xxx_free(...)
xxx_internal_1(...)
...
xxx_internal_m(...)
```

class modules

Each module contains:

- a function `xxx_init(...)` filling the structure `xx`
- a function `xxx_free(...)` freeing all the memory allocated to this structure
- some functions `xxx_external_1(...)`, ..., `xxx_external_n(...)` that can be called from other modules (e.g. to read correctly or interpolate the content of the structure `xx`)
- some functions `xxx_internal_1(...)`, ..., `xxx_internal_m(...)` that are called only inside the module, within `xxx_init(...)`

Following order always respected in `xxx.c`:

```
xxx_external_1(...)
...
xxx_external_n(...)
xxx_init(...)
xxx_free(...)
xxx_internal_1(...)
...
xxx_internal_m(...)
```

Remark: a module in the CLASS code is very similar to a "class" in C++. We enjoy the structure of C++ and the speed of C.

class modules

Following order always respected in `xxx.c`:

```
xxx_external_1(...)
...
xxx_external_n(...)
xxx_init(...)
xxx_free(...)
xxx_internal_1(...)
...
xxx_internal_m(...)
```

Count number of external and internal functions in `source/background.c`:
Search "`int background_`" starting from top

class main

The `main()` function of CLASS located in `main/class.c` could only contains:

```
int main() {
 input_init_..(..,ppr,pba,pth,ppt,ptr,ppm,psp,pnl,ple,pop);
 background_init(ppr,pba);
 thermodynamics_init(ppr,pba,pth);
 perturb_init(ppr,pba,pth,ppt);
 primordial_init(ppr,ppt,ppm);
 nonlinear_init(ppr,pba,pth,ppt,ppm,pnl);
 transfer_init(ppr,pba,pth,ppt,pnl,ptr);
 spectra_init(ppr,pba,ppt,ppm,pnl,ptr,psp);
 lensing_init(ppr,ppt,psp,pnl,ple);
 output_init(pba,pth,ppt,ppm,ptr,psp,pnl,ple,pop)
 /* all calculations done, free the structures */
 lensing_free(ple);
 spectra_free(psp);
 transfer_free(ptr);
 nonlinear_free(pnl);
 primordial_free(ppm);
 perturb_free(ppt);
 thermodynamics_free(pth);
 background_free(pba);
}
```