



INSTITUTO DE
CIENCIAS
FÍSICAS

CLASS: Cosmic Linear Anisotropy Solving System

A short introduction to class

Gabriela García Arroyo (arroyo@icf.unam.mx)

ICF-UNAM

Contents

1 Referencias

2 Primeras pruebas

Referencias principales



<http://class-code.net>



https://github.com/lesgourg/class_public

Main papers

- CLASS I: Overview, by J. Lesgourgues, arXiv:1104.2932 [astro-ph.IM]
- CLASS II: Approximation schemes, by D. Blas, J. Lesgourgues, T. Tram, JCAP 1107 (2011) 034
- CLASS III: Comparison with CAMB for LambdaCDM, by J. Lesgourgues, arXiv:1104.2934 [astro-ph.CO]
- CLASS IV: Efficient implementation of non-cold relics, by J. Lesgourgues, T. Tram, JCAP 1109 (2011) 032

Instalación, desde la terminal

- `$ mkdir Projects/rutafav`
- `$ cd Projects`
- `$ git clone https://github.com/lesgourg/class_public.git`

Instalación, desde la terminal

- `$ mkdir Projects/rutafav`
- `$ cd Projects`
- `$ git clone https://github.com/lesgourg/class_public.git`
- Antes de instalar
- Uso de conda -recomendado si planeas tener más de una versión de class-
 - `$ conda create -name cosmo python=3.xx numpy scipy matplotlib jupyter Cython etc`
 - `$ conda env create -f cosmo.yml` (no necesario si creas tu propio env)
 - `$ conda activate cosmo`
 - `$ cd class/`
 - `$ make clean`
 - `$ make all`
- Wait for errors, talk about your model...

Test your installation

Test your installation

Desde la terminal:

Test your installation

Desde la terminal:

```
■ class]$ python »> import classy
»> cosmo = classy.Class()
»> cosmo.set_baseline('planck2018')
»> cosmo.compute()
»> exit()
```

Test your installation

Desde la terminal:

```
■ class]$ python »> import classy
»> cosmo = classy.Class()
»> cosmo.set_baseline('planck2018')
»> cosmo.compute()
»> exit()
```

Recomiendo -de todas formas-

Test your installation

Desde la terminal:

```
■ class]$ python »> import classy
»> cosmo = classy.Class()
»> cosmo.set_baseline('planck2018')
»> cosmo.compute()
»> exit()
```

Recomiendo -de todas formas-

```
■ class]$ cd notebooks
```

Test your installation

Desde la terminal:

```
■ class]$ python »> import classy
»> cosmo = classy.Class()
»> cosmo.set_baseline('planck2018')
»> cosmo.compute()
»> exit()
```

Recomiendo -de todas formas-

```
■ class]$ cd notebooks
■ notebooks]$ jupyter notebook
```

Test your installation

Desde la terminal:

```
■ class]$ python »> import classy
»> cosmo = classy.Class()
»> cosmo.set_baseline('planck2018')
»> cosmo.compute()
»> exit()
```

Recomiendo -de todas formas-

- class]\$ cd notebooks
- notebooks]\$ jupyter notebook
- Abre el que quieras y solo haz Shift+Enter

Test your installation

Desde la terminal:

```
■ class]$ python »> import classy
»> cosmo = classy.Class()
»> cosmo.set_baseline('planck2018')
»> cosmo.compute()
»> exit()
```

Recomiendo -de todas formas-

- class]\$ cd notebooks
- notebooks]\$ jupyter notebook
- Abre el que quieras y solo haz Shift+Enter

Wait for more errors?

Ejecución: as a 'blackbox'

Desde la terminal:

- `class]$ ./class explanatory.ini`

Ejecución: as a 'blackbox'

Desde la terminal:

- `class]$ ./class explanatory.ini`
- Let's see explanatory.ini

Ejecución: as a 'blackbox'

Desde la terminal:

- `class]$ ./class explanatory.ini`
- Let's see explanatory.ini
- Ejercicio: Indica que quieres un modelo de energía oscura con $w_0 \neq -1$

Ejecución: as a 'blackbox'

Desde la terminal:

- `class]$ ./class explanatory.ini`
- Let's see `explanatory.ini`
- Ejercicio: Indica que quieres un modelo de energía oscura con $w_0 \neq -1$
- Revisemos `/output/files.dat`

Ejecución: as a 'blackbox'

Desde la terminal:

- `class]$ ./class explanatory.ini`
- Let's see `explanatory.ini`
- Ejercicio: Indica que quieres un modelo de energía oscura con $w_0 \neq -1$
- Revisemos `/output/files.dat`
- Puedes usar tu graficador favorito para tu cantidad de interés.

Ejecución: as a 'blackbox'

Desde la terminal:

- `class]$ ./class explanatory.ini`
- Let's see `explanatory.ini`
- Ejercicio: Indica que quieres un modelo de energía oscura con $w_0 \neq -1$
- Revisemos `/output/files.dat`
- Puedes usar tu graficador favorito para tu cantidad de interés.

Desde jupyter:

Ejecución: as a 'blackbox'

Desde la terminal:

- `class]$ ./class explanatory.ini`
- Let's see `explanatory.ini`
- Ejercicio: Indica que quieres un modelo de energía oscura con $w_0 \neq -1$
- Revisemos `/output/files.dat`
- Puedes usar tu graficador favorito para tu cantidad de interés.

Desde jupyter:

- `$ jupyter notebook`

Ejecución: as a 'blackbox'

Desde la terminal:

- `class]$ ./class explanatory.ini`
- Let's see `explanatory.ini`
- Ejercicio: Indica que quieres un modelo de energía oscura con $w_0 \neq -1$
- Revisemos `/output/files.dat`
- Puedes usar tu graficador favorito para tu cantidad de interés.

Desde jupyter:

- `$ jupyter notebook`
- Nos movemos a `jupyter` -por tiempo-

Notebook -updated-



notebook -actualizado-

Modificar class

- 1 De verdad lo necesitas modificar?
- 2 Be sure you need to do it!!

Modificar class

- 1 De verdad lo necesitas modificar?
- 2 Be sure you need to do it!!
- 3 Revisa el montón de beyond LCDM models que ya vienen implementados -explanatori.ini-

Modificar class

- 1 De verdad lo necesitas modificar?
- 2 Be sure you need to do it!!
- 3 Revisa el montón de beyond LCDM models que ya vienen implementados -explanatori.ini-
- 4 Piensa en la forma más fácil -similar- a lo ya existente para que tengas una guía.

Modificar class

- 1 De verdad lo necesitas modificar?
- 2 Be sure you need to do it!!
- 3 Revisa el montón de beyond LCDM models que ya vienen implementados -explanatori.ini-
- 4 Piensa en la forma más fácil -similar- a lo ya existente para que tengas una guía.
- 5 Asegúrate de tener las ecuaciones de tu modelo: ecuación de evolución,

Modificar class

- 1 De verdad lo necesitas modificar?
- 2 Be sure you need to do it!!
- 3 Revisa el montón de beyond LCDM models que ya vienen implementados -explanatori.ini-
- 4 Piensa en la forma más fácil -similar- a lo ya existente para que tengas una guía.
- 5 Asegúrate de tener las ecuaciones de tu modelo: ecuación de evolución, o Hubble,

Modificar class

- 1 De verdad lo necesitas modificar?
- 2 Be sure you need to do it!!
- 3 Revisa el montón de beyond LCDM models que ya vienen implementados -explanatori.ini-
- 4 Piensa en la forma más fácil -similar- a lo ya existente para que tengas una guía.
- 5 Asegúrate de tener las ecuaciones de tu modelo: ecuación de evolución, o Hubble, perturbaciones lineales en norma síncrona o newtoniana,

Modificar class

- 1 De verdad lo necesitas modificar?
- 2 Be sure you need to do it!!
- 3 Revisa el montón de beyond LCDM models que ya vienen implementados -explanatori.ini-
- 4 Piensa en la forma más fácil -similar- a lo ya existente para que tengas una guía.
- 5 Asegúrate de tener las ecuaciones de tu modelo: ecuación de evolución, o Hubble, perturbaciones lineales en norma síncrona o newtoniana, modifica el gauge? -en este caso asegúrate de conocer la transformación-,

Modificar class

- 1 De verdad lo necesitas modificar?
- 2 Be sure you need to do it!!
- 3 Revisa el montón de beyond LCDM models que ya vienen implementados -explanatori.ini-
- 4 Piensa en la forma más fácil -similar- a lo ya existente para que tengas una guía.
- 5 Asegúrate de tener las ecuaciones de tu modelo: ecuación de evolución, o Hubble, perturbaciones lineales en norma síncrona o newtoniana, modifica el gauge? -en este caso asegúrate de conocer la transformación-, Es un campo escalar?

Modificar class

- 1 De verdad lo necesitas modificar?
- 2 Be sure you need to do it!!
- 3 Revisa el montón de beyond LCDM models que ya vienen implementados -explanatori.ini-
- 4 Piensa en la forma más fácil -similar- a lo ya existente para que tengas una guía.
- 5 Asegúrate de tener las ecuaciones de tu modelo: ecuación de evolución, o Hubble, perturbaciones lineales en norma síncrona o newtoniana, modifica el gauge? -en este caso asegúrate de conocer la transformación-, Es un campo escalar? necesitas el potencial y sus primeras dos derivadas -ten cuidado con las unidades-.

Modificar class

- 1 De verdad lo necesitas modificar?
- 2 Be sure you need to do it!!
- 3 Revisa el montón de beyond LCDM models que ya vienen implementados -explanatori.ini-
- 4 Piensa en la forma más fácil -similar- a lo ya existente para que tengas una guía.
- 5 Asegúrate de tener las ecuaciones de tu modelo: ecuación de evolución, o Hubble, perturbaciones lineales en norma síncrona o newtoniana, modifica el gauge? -en este caso asegúrate de conocer la transformación-, Es un campo escalar? necesitas el potencial y sus primeras dos derivadas -ten cuidado con las unidades-.
- 6 Ejemplo simple:

$$\omega_{de} = \omega_0 + \omega_a(1 - a)^7 \quad (1)$$

$$(2)$$

Modificar class

- 1 De verdad lo necesitas modificar?
- 2 Be sure you need to do it!!
- 3 Revisa el montón de beyond LCDM models que ya vienen implementados -explanatori.ini-
- 4 Piensa en la forma más fácil -similar- a lo ya existente para que tengas una guía.
- 5 Asegúrate de tener las ecuaciones de tu modelo: ecuación de evolución, o Hubble, perturbaciones lineales en norma síncrona o newtoniana, modifica el gauge? -en este caso asegúrate de conocer la transformación-, Es un campo escalar? necesitas el potencial y sus primeras dos derivadas -ten cuidado con las unidades-.
- 6 Ejemplo simple:

$$\omega_{de} = \omega_0 + \omega_a(1 - a)^7 \quad (1)$$

$$(2)$$

Seguro puedo aprovechar la implementación de CPL!! :)

- Las ecuaciones viven en: `../class_public/source/archivos.c`

- Las ecuaciones viven en: `../class_public/source/archivos.c`
- Tu modificación afecta al background?

- Las ecuaciones viven en: `../class_public/source/archivos.c`
- Tu modificación afecta al background?
- Ve a `../class_public/source/background.c`

- Las ecuaciones viven en: `../class_public/source/archivos.c`
- Tu modificación afecta al background?
- Ve a `../class_public/source/background.c`
- Afecta a las perturbaciones?

- Las ecuaciones viven en: `../class_public/source/archivos.c`
- Tu modificación afecta al background?
- Ve a `../class_public/source/background.c`
- Afecta a las perturbaciones?
- Ve a `../class_public/source/perturbations.c`

- Las ecuaciones viven en: `../class_public/source/archivos.c`
- Tu modificación afecta al background?
- Ve a `../class_public/source/background.c`
- Afecta a las perturbaciones?
- Ve a `../class_public/source/perturbations.c`
- Modifica inflación?

- Las ecuaciones viven en: `../class_public/source/archivos.c`
- Tu modificación afecta al background?
- Ve a `../class_public/source/background.c`
- Afecta a las perturbaciones?
- Ve a `../class_public/source/perturbations.c`
- Modifica inflación?
- Vamos a `../class_public/source/primordial.c`

- Las ecuaciones viven en: `../class_public/source/archivos.c`
- Tu modificación afecta al background?
- Ve a `../class_public/source/background.c`
- Afecta a las perturbaciones?
- Ve a `../class_public/source/perturbations.c`
- Modifica inflación?
- Vamos a `../class_public/source/primordial.c`

Sin importar que módulo afecte, tienes que revisar
`../class_public/source/input.c`

- Las ecuaciones viven en: `../class_public/source/archivos.c`
- Tu modificación afecta al background?
- Ve a `../class_public/source/background.c`
- Afecta a las perturbaciones?
- Ve a `../class_public/source/perturbations.c`
- Modifica inflación?
- Vamos a `../class_public/source/primordial.c`

Sin importar que módulo afecte, tienes que revisar

`../class_public/source/input.c`

Si modificaste un archivo.c, busca su correspondiente

`../class_public/include/archivo.h` -es altamente probable que lo tengas que modificar-

- Las ecuaciones viven en: `../class_public/source/archivos.c`
- Tu modificación afecta al background?
- Ve a `../class_public/source/background.c`
- Afecta a las perturbaciones?
- Ve a `../class_public/source/perturbations.c`
- Modifica inflación?
- Vamos a `../class_public/source/primordial.c`

Sin importar que módulo afecte, tienes que revisar

`../class_public/source/input.c`

Si modificaste un archivo.c, busca su correspondiente

`../class_public/include/archivo.h` -es altamente probable que lo tengas que modificar-

Recomiendo también darle un vistazo a `cclassy.pxd` y definir el parámetro nuevo.

Ejemplo 1: 7CPL

Ecuaciones principales:

$$\omega_{de} = \omega_0 + \omega_a(1 - a)^7 \quad (3)$$

$$\rho_{de} = \rho_{de}(a) = \rho_{de,0} a^{-3(1+w_0+w_a)} \exp \left[-3w_a \left(\frac{363}{140} - 7a + \frac{21}{2}a^2 - \frac{35}{3}a^3 + \frac{35}{4}a^4 - \frac{21}{5}a^5 + \frac{7}{6}a^6 - \frac{1}{7}a^7 \right) \right] \quad (4)$$

Ejemplo 1: 7CPL

Ecuaciones principales:

$$\omega_{de} = \omega_0 + \omega_a(1 - a)^7 \quad (3)$$

$$\rho_{de} = \rho_{de}(a) = \rho_{de,0} a^{-3(1+w_0+w_a)} \exp \left[-3w_a \left(\frac{363}{140} - 7a + \frac{21}{2}a^2 - \frac{35}{3}a^3 + \frac{35}{4}a^4 - \frac{21}{5}a^5 + \frac{7}{6}a^6 - \frac{1}{7}a^7 \right) \right] \quad (4)$$

Las perturbaciones en class se calculan usando el formalismo ppf (http://class-code.net/PPF_formalism.pdf)

Ejemplo 1: 7CPL

Ecuaciones principales:

$$\omega_{de} = \omega_0 + \omega_a(1 - a)^7 \quad (3)$$

$$\rho_{de} = \rho_{de}(a) = \rho_{de,0} a^{-3(1+w_0+w_a)} \exp \left[-3w_a \left(\frac{363}{140} - 7a + \frac{21}{2}a^2 - \frac{35}{3}a^3 + \frac{35}{4}a^4 - \frac{21}{5}a^5 + \frac{7}{6}a^6 - \frac{1}{7}a^7 \right) \right] \quad (4)$$

Las perturbaciones en class se calculan usando el formalismo ppf (http://class-code.net/PPF_formalism.pdf)

En este caso, no necesitamos modificarlo.

Ejemplo 1: 7CPL

Ecuaciones principales:

$$\omega_{de} = \omega_0 + \omega_a(1 - a)^7 \quad (3)$$

$$\rho_{de} = \rho_{de,0} a^{-3(1+w_0+w_a)} \exp \left[-3w_a \left(\frac{363}{140} - 7a + \frac{21}{2}a^2 - \frac{35}{3}a^3 + \frac{35}{4}a^4 - \frac{21}{5}a^5 + \frac{7}{6}a^6 - \frac{1}{7}a^7 \right) \right] \quad (4)$$

Las perturbaciones en class se calculan usando el formalismo ppf (http://class-code.net/PPF_formalism.pdf)

En este caso, no necesitamos modificarlo.

Prendemos y apagamos las perturbaciones de energía oscura con:
`use_ppf = 'yes'`

Ejemplo2: Inflación

Ejemplo2: Inflación

■ Partamos de explanatory.ini

```

857 # -----
858 # ----> Primordial parameters:
859 # -----
860
861 # 1) Primordial spectrum type
862 #   - 'analytic_Pk' for an analytic smooth function with amplitude, tilt,
863 #     running, etc.; analytic spectra with feature can also be added as
864 #     a new type;
865 #   - 'Inflation_V' for a numerical computation of the inflationary
866 #     primordial spectrum, through a full integration of the perturbation
867 #     equations, given a parametrization of the potential V(phi) in the
868 #     observable window, like in astro-ph/0703625;
869 #   - 'inflation_H' for the same, but given a parametrization of the
870 #     potential H(phi) in the observable window, like in
871 #     astro-ph/0710.1630;
872 #   - 'Inflation_V_end' for the same, but given a parametrization of the
873 #     potential V(phi) in the whole region between the observable part and
874 #     the end of inflation;
875 #   - 'two scales' allows to specify two amplitudes instead of one
876 #     amplitude and one tilt, like in the isocurvature mode analysis of the
877 #     Planck inflation paper (works also for adiabatic mode only; see
878 #     details below, item 2.c);
879 #   - 'external_Pk' allows for the primordial spectrum to be computed
880 #     externally by some piece of code, or to be read from a table, see
881 #     2.d).
882 #   (default: set to 'analytic_Pk')
883 Pk_ini_type = analytic_Pk
884
885 # 1.a) Pivot scale in Mpc-1 (default: set to 0.05)
886 k_pivot = 0.05
887
888 # 1.b) For type 'analytic_Pk':
889 # 1.b.1) For scalar perturbations
890 #   curvature power spectrum value at pivot scale ('A_s' or
891 #   'ln_A_s_1e10') OR one of 'sigma8' or 'S8' (found by iterations using a shooting
892 #   method). (default: set 'A_s' to 2.100549e-09)
893 A_s = 2.100549e-09
894 ln_A_s_1e10 = 3.04478383
895 sigma8 = 0.824398
896 S8 = 0.837868
897
898 # 1.b.1.1) Adiabatic perturbations:
899 #   tilt at the same scale 'n_s', and tilt running 'alpha_s'

```

primordial.c

Eemplo 1. Modifiquemos el espectro analítico, añadiendo el término:

$$P_{\text{new}}(k) = \frac{A_{\text{new}}}{\sqrt{2\pi} \Delta} \exp \left[-\frac{1}{2} \frac{1}{\Delta^2} \ln^2 \frac{k}{k_{\text{pivot}}} \right].$$

primordial.c

Eemplo 1. Modifiquemos el espectro analítico, añadiendo el término:

$$P_{\text{new}}(k) = \frac{A_{\text{new}}}{\sqrt{2\pi} \Delta} \exp \left[-\frac{1}{2} \frac{1}{\Delta^2} \ln^2 \frac{k}{k_{\text{pivot}}} \right].$$

- Tenemos dos parámetros nuevos (agregarlos a primordial.h y a input.c)

primordial.c

Eemplo 1. Modifiquemos el espectro analítico, añadiendo el término:

$$P_{\text{new}}(k) = \frac{A_{\text{new}}}{\sqrt{2\pi} \Delta} \exp \left[-\frac{1}{2} \frac{1}{\Delta^2} \ln^2 \frac{k}{k_{\text{pivot}}} \right].$$

- Tenemos dos parámetros nuevos (agregarlos a primordial.h y a input.c)
- Modificamos primordial.c

primordial.c

Eemplo 1. Modifiquemos el espectro analítico, añadiendo el término:

$$P_{\text{new}}(k) = \frac{A_{\text{new}}}{\sqrt{2\pi} \Delta} \exp \left[-\frac{1}{2} \frac{1}{\Delta^2} \ln^2 \frac{k}{k_{\text{pivot}}} \right].$$

- Tenemos dos parámetros nuevos (agregarlos a primordial.h y a input.c)
- Modificamos primordial.c

```

922 /**
923  * This routine returns the primordial spectrum in the simple analytic case with
924  * amplitudes, tilts, runnings, for each mode (scalar/tensor...),
925  * pair of initial conditions, and wavenumber.
926  *
927  * @param ppm          Input/output: pointer to primordial structure
928  * @param index_md      Input: index of mode (scalar, tensor, ...)
929  * @param index_ic1_ic2 Input: pair of initial conditions (ic1, ic2)
930  * @param k             Input: wavenumber in same units as pivot scale, i.e. in 1/Mpc
931  * @param pk           Output: primordial power spectrum A (k/k_pivot)^(n+...)
932  * @return the error status
933  */
934
935 int primordial_analytic_spectrum(
936     struct primordial * ppm,
937     int index_md,
938     int index_ic1_ic2,
939     double k,
940     double * pk
941 ) {
942
943     if (ppm->is_non_zero[index_md][index_ic1_ic2] == _TRUE_) {
944         *pk = ppm->amplitude[index_md][index_ic1_ic2]
945             * exp((ppm->tltt[index_md][index_ic1_ic2]-1.)*log(k/ppm->k_pivot)
946                 + 0.5 * ppm->running[index_md][index_ic1_ic2] * pow(log(k/ppm->k_pivot), 2.)) /*Modification*/;
947     }
948 }

```

Compila y prueba tu instalación

Desde la terminal:

Compila y prueba tu instalación

Desde la terminal:

- `conda activate cosmo` (Si tienes ambiente virtual)

Compila y prueba tu instalación

Desde la terminal:

- `conda activate cosmo` (Si tienes ambiente virtual)
- `class_public]$ make clean`

Compila y prueba tu instalación

Desde la terminal:

- `conda activate cosmo` (Si tienes ambiente virtual)
- `class_public]$ make clean`
- `class_public]$ make all`

Compila y prueba tu instalación

Desde la terminal:

- `conda activate cosmo` (Si tienes ambiente virtual)
- `class_public]$ make clean`
- `class_public]$ make all`
- Agrega a `explanatory.ini` tus parámetros

Compila y prueba tu instalación

Desde la terminal:

- `conda activate cosmo` (Si tienes ambiente virtual)
- `class_public]$ make clean`
- `class_public]$ make all`
- Agrega a `explanatory.ini` tus parámetros
- No olvides: `write_primordial = yes` (no by default)

Compila y prueba tu instalación

Desde la terminal:

- `conda activate cosmo` (Si tienes ambiente virtual)
- `class_public]$ make clean`
- `class_public]$ make all`
- Agrega a `explanatory.ini` tus parámetros
- No olvides: `write_primordial = yes` (no by default)
- `class_public]$ ./class explanatory.ini`

Compila y prueba tu instalación

Desde la terminal:

- `conda activate cosmo` (Si tienes ambiente virtual)
- `class_public]$ make clean`
- `class_public]$ make all`
- Agrega a `explanatory.ini` tus parámetros
- No olvides: `write_primordial = yes` (no by default)
- `class_public]$ ./class explanatory.ini`
- Si lo deseas, prueba tu modificación en `.ipynb -help(cosmo)-`

Compila y prueba tu instalación

Desde la terminal:

- `conda activate cosmo` (Si tienes ambiente virtual)
- `class_public]$ make clean`
- `class_public]$ make all`
- Agrega a `explanatory.ini` tus parámetros
- No olvides: `write_primordial = yes` (no by default)
- `class_public]$ ./class explanatory.ini`
- Si lo deseas, prueba tu modificación en `.ipynb -help(cosmo)-`
- Let's plot

Perturbaciones

Perturbaciones

- `source/perturbations.c`

Perturbaciones

- `source/perturbations.c`
- Ver paper: Cosmological Perturbation Theory in the Synchronous and Conformal Newtonian Gauges (Ma.Bertschinger)

Perturbaciones

- source/perturbations.c
- Ver paper: Cosmological Perturbation Theory in the Synchronous and Conformal Newtonian Gauges (Ma.Bertschinger)
- CDM en norma síncrona:

$$\delta'_{\text{cdm}} = -\frac{h'}{2},$$

Perturbaciones

- source/perturbations.c
- Ver paper: Cosmological Perturbation Theory in the Synchronous and Conformal Newtonian Gauges (Ma.Bertschinger)
- CDM en norma síncrona:

$$\delta'_{\text{cdm}} = -\frac{h'}{2},$$

- CDM en norma Newtoniana:

$$\delta'_{\text{cdm}} = -\theta_{\text{cdm}} + 3\Phi',$$

$$\theta'_{\text{cdm}} = -\mathcal{H}\theta_{\text{cdm}} + k^2\Psi,$$

```

9210
9217  /** - ---> cdm */
9218
9219  if (pba->has_cdm == _TRUE_) {
9220
9221      /** - ---> newtonian gauge: cdm density and velocity */
9222
9223      if (ppt->gauge == newtonian) {
9224          dy[pv->index_pt_delta_cdm] = -(y[pv->index_pt_theta_cdm]+metric_continuity); /* cdm density */
9225
9226          dy[pv->index_pt_theta_cdm] = - a_prime_over_a*y[pv->index_pt_theta_cdm] + metric_euler; /* cdm veloc
9227      }
9228
9229      /** - ---> synchronous gauge: cdm density only (velocity set to zero by definition of the gauge) */
9230
9231      if (ppt->gauge == synchronous) {
9232          dy[pv->index_pt_delta_cdm] = -metric_continuity; /* cdm density */
9233      }
9234  }

```