Class Diagram for Density Splits Code

Last edited: March 5th, 2019

Class Universe

private:

```
int expansion_or_collapse;
int number_of_time_steps;

double t_initial;
double eta_initial;
double H_initial;
double a_initial;
double a_final;

vector<double> t;
vector<double> eta;
vector<double> a;
vector<double> H;
vector<double> H_prime;

cosmological_model cosmology;
```

public:

```
Universe(cosmological_model cosmo, double a_min, double a_max, int expand_or_collapse);
~Universe();

double return_a_initial();
double return_a_final();
double return_eta_initial();
double return_eta_final();
cosmological_model return_cosmology();
void print_background_cosmology(string filename);

double f_k(double w);
double t_at_eta(double e);
double a_at_eta(double e);
double H_at_eta(double e);
double H_prime_at_eta(double e);
double eta_at_a(double a);
vector<vector<double> > return_background_expansion();
vector<vector<double> > return_background_expansion(int conformal_time_steps);

double rho_m_of_a(double scale); // All in units of TODAYS critical density
double rho_r_of_a(double scale);
double rho_L_of_a(double scale);
double w_L_of_a(double scale);

static void expansion_in_flat_matter_dominated_universe(double a, double *t_phys, double *eta, double *H_conformal, double *H_conformal_prime);
static void expansion_in_flat_radiation_dominated_universe(double a, double *t_phys, double *eta, double *H_conformal, double *H_conformal_prime);
static void expansion_in_flat_Lambda_dominated_universe(double a, double *t_phys, double *eta, double *H_conformal, double *H_conformal_prime);
```

private:

```
void set_initial_conditions();
void set_background_cosmology();
void set_number_of_time_steps(int n_entries);

double hubble_from_Friedmann(double a_start);
double hubble_prime_from_Friedmann(double a_start);
```

struct cosmological_model

```
int collapse;

double Omega_m;
double Omega_r;
double Omega_L;
double Omega_b;
double Omega_k;

double theta_27;
double w0;
double w1;
double n_s;
double h_100;
double sigma_8;
```