

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

Scuola di Scienze
Dipartimento di Fisica e Astronomia
Corso di Laurea in Fisica

Sviluppo ed ottimizzazione delle librerie C++ e Python CosmoBolognaLib

Relatore:
Prof. Federico Marulli

Presentata da:
Luca Pivetti

Anno Accademico 2022/2023

Sogna, credi e realizza.

Abstract

La cosmologia moderna è un campo di studio che si occupa dell'origine, dell'evoluzione e della struttura dell'Universo nel suo complesso, utilizzando osservazioni astronomiche, dati sperimentali e modelli teorici. I problemi tuttora aperti nella cosmologia moderna non sono pochi. Tra i più discussi, c'è la natura e l'origine dell'energia oscura, che sembra guidare l'accelerazione dell'espansione dell'Universo. Un altro problema è la natura e la distribuzione della materia oscura, ovvero una forma di materia invisibile che influisce sulla struttura a grande scala dell'Universo. Inoltre, la comprensione dell'origine dell'Universo stesso, inclusa la sua fase iniziale di espansione accelerata chiamata inflazione cosmica, è ancora un mistero tutto da risolvere.

Questo lavoro di Tesi è dedicato a una parziale ristrutturazione delle librerie di calcolo `CosmoBolognaLib` (CBL), sviluppate presso il Dipartimento di Fisica ed Astronomia dell'Università di Bologna. Si tratta di librerie numeriche *free software* in C++ per calcoli cosmologici. È un progetto in continua evoluzione che mira a definire un ambiente numerico comune per le indagini cosmologiche sulla struttura a grande scala dell'Universo. Il mio lavoro si è concentrato sulla classe `Cosmology`, che contiene tutte le funzioni prettamente cosmologiche. Lo scopo è stato quello di creare nuove classi con cui riorganizzare una grossa parte delle funzioni presenti inizialmente nella classe `Cosmology`. Grazie a questo, è stato possibile migliorare la modularità, la riusabilità del codice, la manutenibilità e la facilità di comprensione.

Il capitolo 1 è dedicato all'introduzione della cosmologia moderna.

Il capitolo 2 presenta la descrizione della struttura e dell'utilizzo delle CBL.

Il capitolo 3 è dedicato alle modifiche apportate al codice e ai miglioramenti ottenuti.

Il capitolo 4, in conclusione, riassume il lavoro svolto e quello in programma per terminare questa ristrutturazione.

Indice

1	Introduzione alla cosmologia moderna	7
1.1	La cosmologia	7
1.2	Cenni di Relatività Generale	9
1.3	Modello Λ CDM	11
1.3.1	Espansione dell'Universo	12
1.3.2	Primi istanti dopo il Big Bang	13
1.4	Materia oscura	15
1.4.1	Evidenze sperimentali	15
1.4.2	Possibili ipotesi sulla materia oscura	17
1.5	Energia oscura	18
1.5.1	L'energia del vuoto	18
1.5.2	Campo scalare	19
1.5.3	Gravità modificata	20
1.5.4	Evidenze osservative dell'accelerazione dell'Universo	20
1.6	Tecniche di analisi cosmologiche	21
2	Libreria di calcolo CosmoBolognaLib	23
2.1	Struttura e contenuti	23
2.1.1	Moduli	24
2.1.2	<i>Namespace</i>	25
2.1.3	Classi	26
2.1.4	File hierarchy	26
2.2	Esempio di utilizzo	27
3	Ristrutturazione della classe Cosmology	29
3.1	Situazione iniziale	29
3.2	Strategia utilizzata	30
3.2.1	Costruzione del nuovo header	31
3.2.2	Modifica del <i>.cpp</i> relativo	32
3.2.3	Modifiche delle altre classi presenti	33
3.2.4	Aggiunta a <i>CBL.h</i> e altri file secondari	34
3.2.5	Conversione in Python	34
3.2.6	Scrittura di nuovi esempi C++ e Python	35
3.2.7	Controlli finali	36
3.3	Nuove classi per le CosmoBolognaLib	37

6

INDICE

4 Conclusioni

39

Bibliografia

41

Capitolo 1

Introduzione alla cosmologia moderna

1.1 La cosmologia

La storia dell'umanità è stata segnata da una progressiva transizione dalla mera osservazione del cielo alla rigorosa indagine scientifica. Inizialmente, gli esseri umani contemplavano il firmamento, cercando di interpretare le costellazioni ed i movimenti astrali. Poi con il passare del tempo, iniziarono ad affidarsi sempre di più al metodo scientifico per comprendere e spiegare i fenomeni celesti in modo accurato e quantitativo. Attraverso l'utilizzo di teorie, modelli matematici e strumenti sempre più avanzati, l'uomo passò dall'osservazione astratta ad una vera e propria scienza astronomica, potendo così indagare sempre più in profondità le questioni non ancora comprese.

Attorno al XVII secolo, la comunità scientifica iniziò a considerare l'Universo come un sistema caratterizzato da uno spazio infinitamente esteso, con il supporto di numerose evidenze sperimentali che sembravano dare conferma di ciò, seppur ovviamente indirette. Tuttavia, secondo i principi dell'ottica, uno spazio di dimensioni infinite, contenente un numero infinito di corpi luminosi, darebbe luogo ad un cielo totalmente luminoso. Questo problema è noto come *paradosso di Olbers*, ed oggi siamo in grado di spiegare perché ciò non accade. Ci sono due ragioni principali per cui l'intensità del campo radiativo intergalattico dovuto alla luce stellare è trascurabile: innanzitutto, l'Universo ha un'età finita, così come le sue sorgenti di luce, come stelle e galassie, il che limita la quantità di luce complessivamente prodotta; in secondo luogo, esiste un orizzonte dovuto alla velocità finita della luce che limita le dimensioni dell'Universo osservabile.

La cosmologia moderna si basa su due principi fondamentali:

- **Principio Copernicano:** Non siamo osservatori privilegiati all'interno dell'Universo.
- **Principio Cosmologico:** L'Universo su grande scala ($r > 200$ Mpc) è omogeneo e isotropo.

Nella seguente figura, si possono intuire queste proprietà osservando la ricostruzione della struttura su larga scala dell'Universo, ottenuta tramite una simulazione cosmologica a molti corpi ¹.

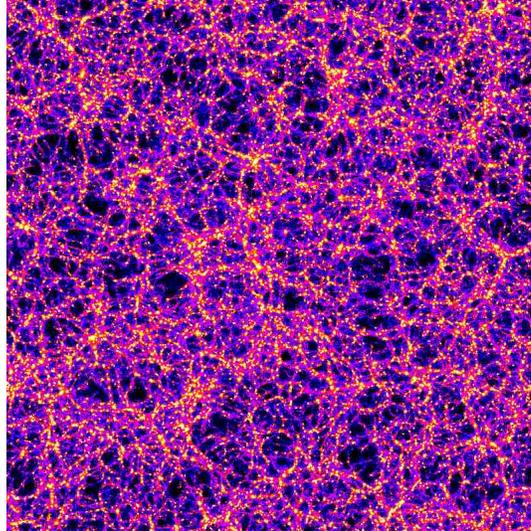


Figura 1.1: Immagine su larga scala della struttura della materia dell'Universo, realizzata con il tool TIPSY. Rappresenta uno spazio di 300 milioni di anniluce cubi, con al suo interno 47 milioni di particelle interagenti tramite forza di gravità. I colori rappresentano la densità di materia presente in ogni punto; al crescere della densità, i colori variano nell'ordine: nero, blu, viola, rosso, arancione, giallo, bianco.

Questi due principi hanno portato a conseguenze profonde sul modo di studiare l'Universo. Grazie ad essi infatti, è stato possibile applicare approssimazioni di simmetria per risolvere le equazioni di campo della Relatività Generale di Einstein, che permettono di descrivere la struttura dello spazio-tempo. A livello matematico, l'isotropia e l'omogeneità dell'Universo sono rappresentate da due triplette di vettori di Killing di tipo spazio: la prima per le tre direzioni di traslazione che descrive l'omogeneità e la seconda per le tre direzioni di rotazione che descrive l'isotropia. È possibile dimostrare che questi sei vettori sono il numero massimo ottenibile per una varietà tridimensionale, portando alla massima simmetria possibile. Aggiungendo poi una dimensione per il tempo t nel costruito matematico, si ottiene la superficie tridimensionale Σ_t che descrive la componente dell'Universo in un determinato istante. Dall'insieme di queste superfici con t differenti, si può ricostruire l'evoluzione completa dell'Universo.

Su questi principi, si basa la *metrica di Friedmann-Lemaître-Robertson-Walker* (FLRW), alla base del modello cosmologico standard. Ponendo la velocità della luce $c = 1$ per comodità di calcolo, questa è definita in coordinate polari (r, θ, ϕ) come segue:

$$ds^2 = -dt^2 + R^2(t) \left[\frac{dr^2}{1 - kr^2} + r^2 (d\theta^2 + \sin^2(\theta)d\phi^2) \right]. \quad (1.1)$$

¹<http://faculty.washington.edu/trq/hpcc/tipsy/picture/picture.html>

Per il Principio Copernicano, l'origine $r = 0$ è arbitraria; t è il tempo cosmico, ovvero il tempo proprio misurato da un osservatore comovente, $R(t)$ è il fattore di scala e k è il parametro di curvatura, con possibili valori $0, \pm 1$.

1.2 Cenni di Relatività Generale

Per una descrizione appropriata della cosmologia moderna è necessaria anche un'introduzione alla teoria della Relatività Generale, sviluppata da Albert Einstein nel 1915 [14]. Prendiamo innanzitutto in considerazione l'equazione di campo di Einstein:

$$G_{\mu\nu} + \Lambda g_{\mu\nu} = 8\pi G T_{\mu\nu}, \quad (1.2)$$

in cui $G_{\mu\nu} \equiv R_{\mu\nu} - \frac{1}{2}g_{\mu\nu}R$ è il tensore di Einstein che esprime la curvatura dello spazio-tempo, $R_{\mu\nu}$ è il tensore di curvatura di Ricci-Curbastro che esprime la curvatura di una varietà riemanniana, $g_{\mu\nu}$ è il tensore metrico che è un campo tensoriale caratterizzante della geometria della varietà, R è la curvatura scalare, coincidente con la traccia di $R_{\mu\nu}$, $T_{\mu\nu}$ è il tensore energia-impulso che descrive il flusso di energia e quantità di moto associato ad un campo di materia, Λ è la costante cosmologica, G è la costante di gravitazione universale.

Il modo più efficace di modellizzare la distribuzione di materia nell'Universo è tramite un fluido omogeneo e isotropo che riempie lo spazio-tempo, in accordo con il principio cosmologico. In questo caso, il suo tensore massa-energia in un sistema di riferimento comovente con l'espansione dello spazio risulta essere

$$T_{\nu}^{\mu} = \text{diag}(-\rho, p, p, p), \quad (1.3)$$

con ρ la densità di energia e p la pressione del fluido. Imponendo le equazioni di continuità e di stato per questo *fluido cosmico*, otteniamo il sistema:

$$\begin{cases} \dot{\rho} + 3H(\rho + p) = 0 \\ p = w\rho \end{cases},$$

dove w è il parametro dell'equazione di stato del fluido e $H = \dot{R}/R$ è il *parametro di Hubble*. Risolvendo, si ottiene la relazione di proporzionalità:

$$\rho \propto R^{-3(1+w)}, \quad (1.4)$$

il cui significato è fondamentale nella cosmologia moderna. Da questa relazione, si definiscono infatti tre tipi di *fluido cosmico*, in base ai valori di w :

- **Materia:** con $w = 0$, e dunque $p = 0$, da cui $\rho_M \propto R^{-3}$.
- **Radiazione:** con $w = 1/3$ e quindi $\rho_R \propto R^{-4}$.
- **Costante cosmologica Λ :** con $w = -1$, da cui si ottiene una densità costante rispetto al fattore di scala: $\rho_{\Lambda} = -p$.

Ora, applicando la metrica FLRW alle equazioni di campo di Einstein, si può ricavare l'equazione di Friedmann

$$H^2 = \frac{8\pi G}{3}\rho - \frac{k}{R^2}, \quad (1.5)$$

in cui si definisce il *parametro di densità* come:

$$\Omega = \frac{8\pi G}{3H^2}\rho = \frac{\rho}{\rho_c}, \quad (1.6)$$

dove $\rho_c = 3H^2/8\pi G$ è la *densità critica dell'Universo*. Si noti che dal valore di questa densità dipende la geometria dell'Universo (nella Figura 1.2 è possibile avere un confronto visivo delle tre geometrie matematicamente possibili):

- $\rho = \rho_c$: in questo caso si ha $\Omega = 1$, $k = 0$ e dunque un **Universo spazialmente piatto**, descrivibile cioè con uno spazio \mathbb{R}^3 . Questo è mostrato nella rappresentazione più in basso della Figura 1.2.
- $\rho < \rho_c$: in questo caso si ha $\Omega < 1$, $k < 0$ e dunque un **Universo spazialmente aperto** e come nel primo caso infinito, ma con la geometria di un'iperbole. Questa è mostrata nella rappresentazione centrale della Figura 1.2.
- $\rho > \rho_c$: in questo caso si ha $\Omega > 1$, $k > 0$ e dunque un **Universo spazialmente chiuso**, cioè caratterizzato dalla geometria di una sfera, come rappresentato in alto nella Figura 1.2.

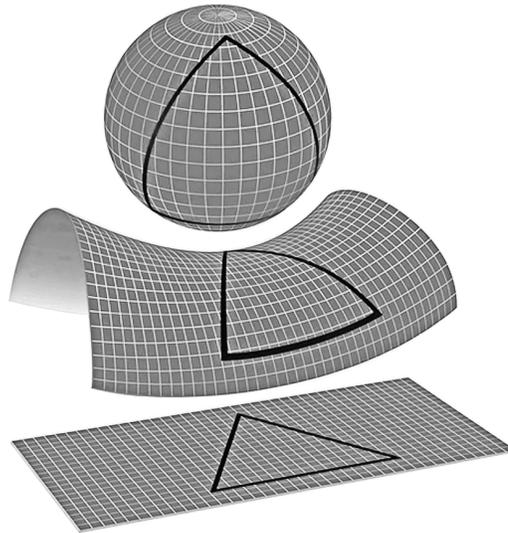


Figura 1.2: Tre rappresentazioni per mostrare le possibili geometrie dell'Universo in base al valore di Ω : nel primo caso abbiamo un Universo chiuso, nel secondo un Universo aperto, nel terzo un Universo piatto [13].

I dati osservativi attuali indicano una densità $\Omega \simeq 1$, [10]. Ricadremmo quindi in un Universo spazialmente piatto.

1.3 Modello Λ CDM

Il modello Λ CDM è attualmente quello che meglio descrive le osservazioni disponibili. Esso si basa su 2 elementi che costituirebbero circa il 95% dell'Universo [9].

Il primo, indicato dalla lettera greca Λ già introdotta in precedenza, è la costante cosmologica e rappresenta l'energia oscura (o del vuoto). Questa energia esercita una repulsione gravitazionale fra ogni corpo presente nello spazio ed è presente ogni istante in ogni posizione. La costante cosmologica era stata inizialmente introdotta da Albert Einstein per cercare di ottenere un Universo statico; tuttavia, assumendo l'uguaglianza fra forza gravitazionale a piccola scala e forza repulsiva, si avrebbe in realtà un equilibrio instabile: ad una minima espansione di questo spazio finito, la densità di materia andrebbe a diminuire, ma essendo la costante cosmologica sempre presente porterebbe la forza repulsiva a vincere ulteriormente su quella attrattiva. Di conseguenza, l'Universo continuerebbe ad espandersi sempre più velocemente. Pochi anni dopo che Einstein sviluppò la Relatività Generale e questi primi modelli cosmologici, Hubble osservò per la prima volta l'espansione dell'Universo notando uno spostamento verso il rosso delle lunghezze d'onda, e la soluzione statica precedentemente proposta da Einstein venne abbandonata, insieme alla costante cosmologica Λ . Venne però in seguito riproposta dopo la formulazione del modello di *Big Bang caldo*, secondo cui l'Universo si sarebbe formato a partire da una singolarità a temperature elevatissime, seguita da un'espansione dello spazio tuttora in corso. A fine XX secolo, due gruppi di ricercatori hanno dimostrato indipendentemente che l'Universo è attualmente in uno stato di espansione accelerata [11], e per spiegare questo, la costante cosmologica Λ tornò in primo piano.

Il secondo termine indicato dall'acronimo *CDM* (Cold Dark Matter), è la materia oscura fredda. Questa è la componente di materia oscura che al tempo del disaccoppiamento tra radiazione e materia (vedi paragrafo 1.3.2) si trovava in uno stato non-relativistico.

Come anticipato, tramite il parametro geometrico di curvatura k , oggi l'Universo sembra avere una geometria spaziale piatta, come precedentemente detto rappresentata dal parametro di densità $\Omega \simeq 1$. Ovviamente la relazione tra questi due parametri è strettissima data la loro dipendenza reciproca. Le più recenti osservazioni hanno permesso di vincolare la densità delle diverse componenti dell'Universo, mostrate in Figura 1.3, che risultano [9]:

- **Materia ordinaria:** $\Omega_M \simeq 0.05$
- **Materia oscura:** $\Omega_{DM} \simeq 0.26$
- **Energia oscura:** $\Omega_{DE} \simeq 0.69$

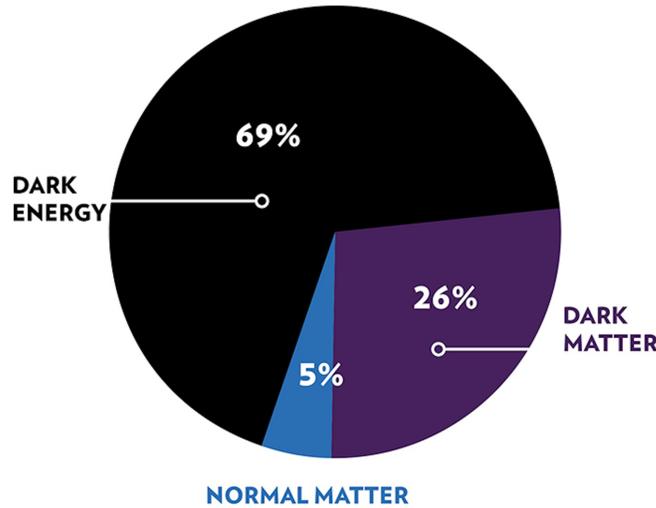


Figura 1.3: Grafico a torta per confrontare la quantità di materia ordinaria rispetto a materia ed energia oscura.

A queste, sarebbe corretto aggiungere anche il contributo della radiazione, ma nel complesso ha un valore trascurabile (attorno allo 0.05%).

1.3.1 Espansione dell'Universo

Il modello dell'espansione dell'Universo ha trovato supporto empirico grazie alle osservazioni di Hubble ad inizio XX secolo, come anticipato nel capitolo precedente. Egli notò infatti che la radiazione emessa dalla maggior parte delle galassie aveva una lunghezza d'onda spostata verso il rosso, e questa aumentava al crescere della distanza della sorgente. Le prime ipotesi riguardarono l'*effetto Doppler* per due corpi in allontanamento, ma ci si rese conto che questo non era sufficiente; da qui, ebbero inizio gli studi che portarono a un'interpretazione della *legge di Hubble-Lemaître* come espansione dello spazio, come predetto dalla Relatività Generale. Una prima formulazione è la seguente:

$$v = H_0 D, \quad (1.7)$$

in cui v è la velocità tra i due corpi considerati, D è la distanza tra di essi e H_0 la *costante di Hubble* al tempo presente. Al crescere della distanza fra due corpi, cresce la velocità di allontanamento fra essi. Mettendo a sistema l'Eq. (1.7) con la legge oraria non relativistica $D = vt$, si ottiene una prima stima riguardante il tempo di vita dell'Universo dal Big Bang fino ad oggi

$$t_H = \frac{1}{H_0} \sim 13 \times 10^9 \text{yr}, \quad (1.8)$$

ed il risultato è in pieno accordo con le più recenti osservazioni sperimentali. Assumendo la velocità di espansione come quella dei fotoni che si allontanano dal primo istante dopo il Big Bang, si ricava facilmente la dimensione raggiunta dallo spazio che compone l'Universo:

$$D_H = c \cdot t_H \sim 4 \text{ Gpc.} \quad (1.9)$$

Andando ora ad analizzare lo spostamento verso il rosso, denominato *redshift*, si può ricavare la *legge di Hubble-Lemaître*:

$$z = \frac{v}{c} = \frac{H_0 D}{c}. \quad (1.10)$$

1.3.2 Primi istanti dopo il Big Bang

L'inizio dell'Universo, secondo la teoria del Big Bang, è il momento primordiale in cui l'intera materia, l'energia e lo spazio-tempo hanno avuto origine. Il Big Bang è un evento cosmologico di eccezionale importanza, che risale a circa 13,8 miliardi di anni fa, in cui tutto l'Universo era racchiuso in un'unica singolarità. Da qui iniziò improvvisamente un'espansione tuttora in corso, che portò alla forma attuale dell'Universo. In realtà, però, tramite la Relatività Generale e le teorie di campo quantistico è possibile ricostruire l'evoluzione fino ad un tempo

$$t_P = c \cdot d_P, \quad (1.11)$$

denominato *tempo di Planck* ($\sim 10^{-43}$ s), ovvero il tempo necessario a un fotone per percorrere una distanza di Planck ($\sim 10^{-35}$ m) (viaggiando alla velocità della luce c).

Di seguito è riportata una descrizione qualitativa delle fasi principali dell'evoluzione dell'Universo, a partire da questo istante.

- **Era di grande unificazione** ($t_P \leq t \leq 10^{-35}$ s): l'intero Universo osservabile ha il diametro di circa 10^{-35} m ed è ad una temperatura di 10^{30} K. A causa delle condizioni presenti, le forze fondamentali dell'Universo sono ancora unite in una sola forza fondamentale.
- **Era dell'inflazione** (10^{-35} s $\leq t \leq 10^{-32}$ s): l'Universo osservabile ha il diametro di circa 10^{-26} m ed è ad una temperatura di 10^{27} K. Questa ipotetica fase fu caratterizzata da una espansione accelerata, in cui si generarono coppie di particelle ed antiparticelle, per poi annichilirsi a vicenda. Tuttavia, probabilmente a causa di una piccola fluttuazione quantistica, ci fu un piccolo eccesso di particelle, responsabile della materia presente nell'Universo attuale.
- **Era elettrodebole** (10^{-32} s $\leq t \leq 10^{-9}$ s): l'Universo osservabile ha il diametro di circa 10 m (si notino i 27 ordini di grandezza di differenza rispetto all'inizio dell'inflazione) e la temperatura è rimasta invariata, anche a causa del brevissimo tempo trascorso. L'interazione forte si è separata dalla restante interazione elettrodebole. Sarà questa interazione a permettere ai nuclei atomici di formarsi.
- **Fine era elettrodebole** (10^{-9} s $\leq t \leq 10^{-6}$ s): l'Universo osservabile ha ora il diametro di circa 10^{12} m e la temperatura è scesa a 10^{15} K. Interazione debole e forza elettromagnetica si separano, conferendo le proprietà di massa a bosoni deboli, quark e leptoni.

- **Era degli adroni** ($10^{-6} \text{ s} \leq t \leq 100 \text{ s}$): l'Universo osservabile ha il diametro di circa 10^{14} m e la temperatura è attorno ai 10^{13} K . A causa dell'energia termica che ha raggiunto un livello sufficientemente basso, i quark iniziano ad interagire con gli antiquark tramite interazione forte, creando i primi adroni.
- **Era della nucleosintesi** ($100 \text{ s} \leq t \leq 200 \text{ s}$): l'Universo osservabile ha un diametro di oltre 10^{15} m e la temperatura è scesa a 10^{10} K . In questa era gran parte dei neutroni decade in protoni e con la temperatura sufficientemente bassa si formarono i primi nuclei di elio-4 e deuterio.
- **Era della radiazione** ($200 \text{ s} \leq t \leq 300\,000 \text{ yr}$): l'Universo osservabile ha un diametro superiore ai 10 anni luce e la temperatura è arrivata a 10^8 K . Con un'energia così bassa, anche l'energia elettromagnetica poté iniziare a dare i suoi effetti, formando i primi atomi di idrogeno con i relativi isotopi, elio e litio.
- **Era della materia** ($t \geq 300\,000 \text{ yr}$): l'Universo osservabile ha un diametro di oltre 10^8 anni luce e la temperatura è scesa fino a 3000 K . I fotoni rimasti dall'inflazione iniziarono a disperdersi, formando la cosiddetta *radiazione cosmica di fondo*, oggi al centro degli studi. In questo periodo iniziò la formazione di stelle e l'espansione dell'Universo, che stava rallentando, in tempi recenti ricominciò ad accelerare.

Di seguito è riportata un'immagine ² che mostra l'espansione dell'Universo fin dai primi istanti dopo il Big Bang, mettendo in evidenza l'estrema rapidità dell'espansione iniziale.

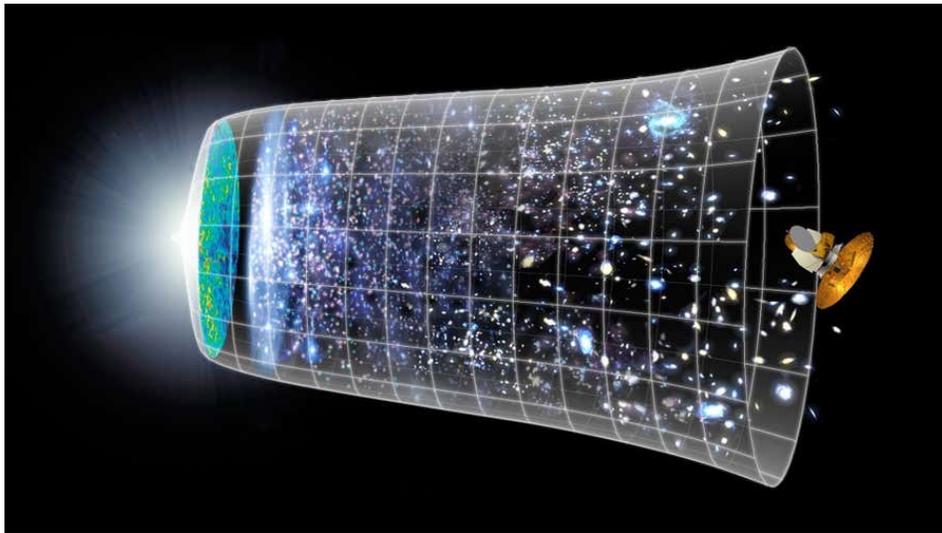


Figura 1.4: In questa immagine è possibile vedere i momenti principali di evoluzione dell'Universo: dalla singolarità iniziale, al periodo di inflazione con il primo grande accrescimento spaziale, al momento di *dark ages*, fino alla formazione delle strutture di materia e in conclusione con i giorni nostri, in cui è stata osservata un'ulteriore accelerazione nell'accrescimento spaziale.

²<https://tech.everyeye.it/articoli/speciale-inflazione-la-solidita-incertezza-50782.html>

1.4 Materia oscura

Come mostrato nel capitolo 1.3, attualmente sembra che solamente un 5% dell'intero Universo sia composto da materia ordinaria. Infatti, numerose osservazioni astronomiche di galassie, tra cui anche la nostra Via Lattea, ammassi di galassie e fenomeni di lensing gravitazionale, hanno mostrato incongruenze tra la massa gravitazionale e quella osservata tramite radiazione elettromagnetica [2]. Questo problema si può risolvere tramite l'introduzione della materia non barionica, detta *materia oscura*. Questa si comporterebbe esattamente come se fosse materia ordinaria per quanto riguarda l'interazione col campo gravitazionale, ma non interagirebbe con il campo elettromagnetico.

1.4.1 Evidenze sperimentali

I primi dati osservativi che hanno suggerito la presenza di materia oscura nell'Universo sono stati ottenuti tramite l'osservazione delle orbite delle stelle vicine al centro delle galassie e delle nubi di gas più esterne. Utilizzando le equazioni della meccanica classica, è possibile determinare l'accelerazione centripeta di un oggetto massivo a distanza r dal centro di una galassia, andante a una velocità v . Approssimando l'orbita con una circonferenza, si ha:

$$a = \frac{v^2}{r}. \quad (1.12)$$

Assumendo che la distribuzione di massa della galassia, $M(r)$, l'accelerazione dovuta alla forza gravitazionale, in approssimazione newtoniana, risulta:

$$a = \frac{GM(r)}{r^2}, \quad (1.13)$$

che a sistema con l'equazione (1.12) permette di ottenere il modulo della velocità del corpo:

$$v = \sqrt{\frac{GM(r)}{r}} \propto \frac{1}{\sqrt{r}}. \quad (1.14)$$

Tramite misure di redshift, si possono ricavare misure estremamente precise della velocità di rotazione di corpi più o meno vicini al centro della galassia. Queste, a differenza di quanto atteso, superata una certa distanza dal centro della galassia, smettono di decrescere come il modello newtoniano appena descritto prevedeva. Questo, dunque, ha indotto gli astronomi ad assumere che la materia presente nelle galassie non è solo quella osservabile tramite radiazione elettromagnetica. Di seguito, come esempio, è riportato un grafico che mostra la differenza tra le curve, attesa e misurata, della velocità di rotazione in funzione del raggio dal centro di una galassia a spirale.

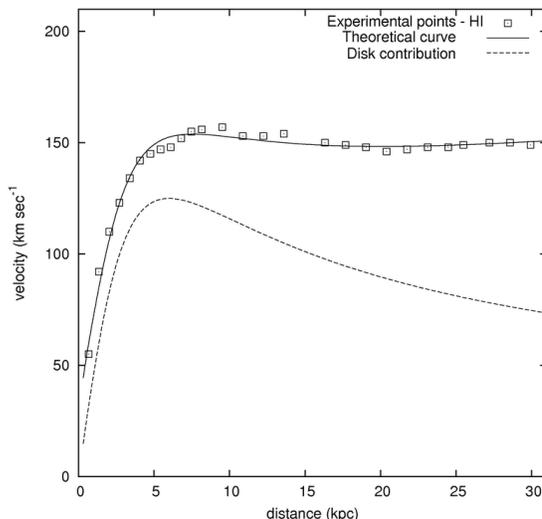


Figura 1.5: Confronto tra i dati osservati (linea superiore) e quelli attesi secondo la meccanica newtoniana (linea inferiore) riguardo le velocità di rotazione della materia presente nel disco della galassia in funzione del suo raggio. Si tratta di osservazioni della galassia a spirale NGC 3198 [3].

Un secondo metodo per misurare la massa di materia oscura presente nell'Universo consiste nell'analisi delle proprietà fisiche degli ammassi di galassie. L'astronomo svizzero Fritz Zwicky per primo misurò la dispersione di velocità all'interno di 7 galassie dell'ammasso della Chioma, e ottenne dei valori non consistenti con quanto predetto, considerando la massa osservata. Assumendo la dinamica newtoniana e la massa osservata, la misura della dispersione delle velocità non erano infatti compatibili con l'ipotesi di orbite stabili [18]. Seguendo la trattazione newtoniana, definiamo l'energia potenziale gravitazionale U all'interno di una galassia come:

$$U = -\alpha \frac{GM^2}{r_H}, \quad (1.15)$$

in cui α è un parametro dipendente dalla distribuzione di densità dell'ammasso, M la massa totale dell'ammasso ed r_H il raggio contenente metà della massa totale dell'ammasso, assumendo che questo abbia forma circolare. Consideriamo l'energia cinetica della galassia, pari a

$$T = \frac{1}{2}M\langle v^2 \rangle, \quad (1.16)$$

in cui $\langle v^2 \rangle$ è la velocità quadratica media della galassie appartenenti all'ammasso, definita come

$$\langle v^2 \rangle \equiv \frac{1}{M} \sum_i m_i |\dot{x}_i|^2. \quad (1.17)$$

Per il teorema viriale, sappiamo che:

$$\langle U \rangle + 2\langle T \rangle = 0, \quad (1.18)$$

e da qui possiamo ricavare

$$M = \frac{\langle v^2 \rangle r_H}{\alpha G}. \quad (1.19)$$

Conoscendo i valori $\langle v^2 \rangle$, α e r_H è possibile quindi derivare la massa dell'ammasso. Ad esempio, nel caso dell'ammasso della Chioma [12], si ottiene:

$$M_{Chioma} \simeq 2 \times 10^{15} M_\odot, \quad (1.20)$$

nonostante la massa misurata di stelle e gas sia pari a circa il 10% di M_{Chioma} .

Anche in questo caso, risulta evidente che la massa necessaria per mantenere stabili le orbite delle stelle nell'ammasso è diversa da quella osservabile.

1.4.2 Possibili ipotesi sulla materia oscura

La grande maggioranza della materia che compone l'Universo si ritiene sia quindi materia oscura, ma ancora non è nota la sua natura fisica e la sua composizione. Una delle sue caratteristiche fondamentali è che non interagisce con i fotoni; per questo motivo, una delle ipotesi più supportate al momento è quella che sia costituita da particelle non barioniche. Attualmente, l'unica particella non barionica di cui è stata dimostrata l'esistenza è il neutrino massivo. Nel *Modello Standard*, solitamente si considera il neutrino come particella di massa nulla. Si può ricavare la seguente relazione tra neutrini e radiazione cosmica di fondo (CMB):

$$n_\nu = 3 \cdot \frac{3}{11} n_\gamma \simeq 3.36 \times 10^8 \text{ m}^{-3}, \quad (1.21)$$

in cui n_ν è la densità dei neutrini, $\frac{3}{11}$ è il rapporto tra il numero di neutrini e quello di fotoni della CMB e n_γ la densità dei fotoni. Utilizzando questo dato si può stimare la massa dei neutrini nell'assunzione che tutta la materia oscura dell'Universo sia composta da queste particelle. Si ottiene:

$$m_\nu = \frac{\Omega_{DM} \rho_c}{n_\nu} \simeq 4 \text{ eV}. \quad (1.22)$$

Tuttavia, questo valore risulta essere troppo alto, in quanto le stime attuali pongono un limite superiore alla massa totale di neutrini presenti nell'Universo pari a circa 0.12 eV, con un livello di confidenza del 95% [9]. Questo è uno dei motivi per cui il modello denominato *Hot Dark Matter* (HDM), in cui i neutrini erano il candidato naturale materia oscura, è oggi considerato non accurato. Secondo questo modello la velocità della materia oscura al momento di disaccoppiamento (nei primi istanti dopo il Big Bang) sarebbe stata relativistica - da cui compare la parola *Hot* nel nome del modello. Inoltre il modello HDM predice una struttura dell'Universo completamente diversa da quella che osserviamo oggi.

Si è così passati alla cosiddetta teoria della *Cold Dark Matter* (CDM), che come detto rappresenta la componente di materia oscura del modello standard della cosmologia moderna. Questo si basa sull'esistenza di particelle non barioniche molto più massive dei neutrini, che al momento del disaccoppiamento hanno mantenuto velocità non relativistiche. Sono stati proposti diversi modelli di nuove particelle come candidati di CDM. In Italia, per esempio, presso i Laboratori Nazionali del Gran Sasso dell'INFN è in corso l'esperimento XENONnT, che indaga la possibile esistenza delle cosiddette *Weakly Interactive Massive Particles* (WIMPs), particelle massive interagenti debolmente. Al momento nessun risultato ha dato dimostrazione della loro esistenza. Un'altra opzione considera particelle denominate assioni, ma anche in questo caso non ci sono ancora prove della loro esistenza.

Un'altra possibilità è che parte della materia oscura sia in realtà semplice materia barionica che non emette i fotoni necessari per essere rilevata. Questa andrebbe a comporre i Massive Compact Halo Objects (MACHOs), *oggetti compatti* come nane bianche, stelle di neutroni e buchi neri, ma anche nane brune e pianeti gioviani. Tuttavia, se anche questo fosse vero, questi oggetti non avrebbero una massa sufficiente per spiegare le proprietà delle strutture a grande scala, che in ogni caso necessiterebbero di materia non barionica.

1.5 Energia oscura

Durante la trattazione dell'evoluzione cosmica, si è spesso parlato di uno spazio in espansione accelerata, tuttora in corso. Ma a cosa è dovuta questa accelerazione? Proprio questa è una delle questioni più discusse oggi nella cosmologia moderna. La teoria più accreditata assume la presenza della cosiddetta *energia oscura*, rappresentata nel modello cosmologico standard dalla costante cosmologica Λ , introdotta nel capitolo 1.3.

1.5.1 L'energia del vuoto

Una prima interpretazione dell'energia oscura è quella di considerarla come una forma di energia presente sempre in ogni punto dello spazio che compone l'Universo. Questa energia non sarebbe quindi associata alla materia o ad un particolare fenomeno in corso. Sarebbe invece una proprietà caratteristica dello spazio stesso. Questo significa che ogni frammento di spazio ha la sua energia - detta anche *energia del vuoto* - e questa sarebbe responsabile di tutti i fenomeni particellari di creazione ed annichilazione particella-antiparticella. Sarebbe dunque questa energia a causare l'espansione accelerata dello spazio.

Assumendo il tensore energia-impulso del vuoto come:

$$T_{\mu\nu} = \rho_{vac} g_{\mu\nu}, \quad (1.23)$$

le cui componenti in coordinate cartesiane sono:

$$\begin{cases} T_{\nu}^{\mu} = \text{diag}(-\rho, p, p, p) \\ g_{\nu}^{\mu} = \delta_{\nu}^{\mu} \end{cases}$$

dove $T^\mu{}_\nu = g^\alpha{}_\mu T_{\alpha\nu}$ e $g^\mu{}_\nu = g^{\mu\alpha} g_{\alpha\nu}$, da cui possiamo ricavare:

$$\rho_{vac} = -p_{vac}, \quad (1.24)$$

confermando l'equivalenza matematica tra energia del vuoto e costante cosmologica.

Conoscendo il valore attuale $\Omega \simeq 1$, possiamo dedurre che oggi la densità dell'Universo è quasi uguale a quella critica:

$$\rho \simeq \rho_c \simeq 10^{-29} g/cm^3 \simeq 10^{-47} \text{ GeV}^4. \quad (1.25)$$

Tuttavia, secondo calcoli approssimativi basati sull'attuale teoria di campo quantistico, questo valore dovrebbe essere molto maggiore:

$$\rho_{vac} \sim 10^{73} \text{ GeV}^4. \quad (1.26)$$

Questo è il motivo per cui oggi si parla di *problema della costante cosmologica*, questione molto discussa oggi nella cosmologia moderna, riguardando una componente fondamentale dell'equazione di Relatività Generale di Albert Einstein.

1.5.2 Campo scalare

Una ipotesi alternativa alla costante cosmologica consiste nell'aggiungere un campo scalare, ϕ , spazialmente omogeneo, come nuovo grado di libertà per rappresentare l'energia oscura. Il tensore energia-impulso coincide con quello di un fluido perfetto secondo la modellizzazione del fluido cosmico, all'interno del quale compaiono i valori

$$\begin{cases} \rho = \frac{\dot{\phi}^2}{2} + V(\phi) \\ p = \frac{\dot{\phi}^2}{2} - V(\phi) \end{cases},$$

dove la prima componente, $\frac{\dot{\phi}^2}{2}$, rappresenta l'energia cinetica e la seconda, $V(\phi)$, rappresenta l'energia potenziale del campo ϕ . A questo punto, si può ricavare il valore adimensionale

$$w = \frac{\frac{\dot{\phi}^2}{2} - 1}{\frac{\dot{\phi}^2}{2} + 1}, \quad (1.27)$$

uguale al rapporto tra pressione p e densità ρ . È ora evidente che, con un'evoluzione lenta di ϕ , ovvero $\frac{\dot{\phi}^2}{2V} \ll 1$, risulta

$$w \approx -1. \quad (1.28)$$

In questo modo questo campo scalare si comporta come un'energia del vuoto, con $\rho_{vac} \simeq V[\phi(t)]$. Si può inoltre notare come il valore di questa energia possa variare solo da -1 a +1.

Tuttavia, il problema della costante cosmologica rimane presente. Si può comunque concludere che, secondo questo modello, l'accelerazione cosmica è dovuta al fatto che l'Universo non ha ancora raggiunto il livello di vuoto stabile.

1.5.3 Gravità modificata

Un'altra possibile ipotesi per spiegare l'espansione accelerata dell'Universo è quella di considerare modifiche alle equazioni di campo della Relatività Generale. Secondo questa teoria, infatti, l'espansione accelerata potrebbe indicare la necessità di modificare le equazioni che descrivono l'evoluzione del tasso di crescita delle perturbazioni, pur tenendo inalterate le equazioni di Friedmann [4].

È stata anche considerata la possibilità di un Universo non omogeneo, mantenendo invariate le equazioni della Relatività Generale. Tuttavia, la maggior parte della comunità scientifica non considera credibile questa soluzione [6] [1].

1.5.4 Evidenze osservative dell'accelerazione dell'Universo

Come abbiamo visto, la comunità scientifica ha sviluppato diversi modelli di energia oscura. È dunque fondamentale confrontare questi modelli con i dati osservativi disponibili, per valutarne l'accuratezza. In particolare, risulta conveniente misurare il parametro w , definito nel capitolo 1.5.2 come rapporto tra la pressione e la densità di energia oscura, che di conseguenza fornisce un buon collegamento con la realtà fenomenologica. La parametrizzazione più semplice è $w = -1$ costante, che insieme a Ω_{DE} fornisce una buona descrizione delle osservazioni disponibili.

Utilizzando questi due parametri, w e Ω_{DE} , sono possibili diverse descrizioni; quasi tutte forniscono risultati consistenti a z bassi, ma molte perdono il senso fisico a z alti, con valori $w \ll -1$ o a $w \gg 1$. Esiste una parametrizzazione, suggerita da [7], che evita questo problema assumendo la presenza di un campo scalare dinamico (non più costante, come descritto nel capitolo (1.5.2)):

$$w(a) = w_0 + w_a(1 - a) = w_0 + \frac{w_a z}{1 + z}, \quad (1.29)$$

in cui w_0 è il valore attuale di w e w_a è la derivata negativa di w rispetto al fattore di scala. Nel caso della costante cosmologica abbiamo $w_0 = -1$ e $w_a = 0$, in buon accordo con i dati osservativi disponibili.

Recenti vincoli osservativi sull'andamento di $w(z)$ sono rappresentati in Figura 1.6. Inoltre, la descrizione a due parametri di $w(z)$, lineari nei parametri, comporta l'esistenza di un redshift "pivot" z_p al quale le misure dei due parametri non sono correlate e di conseguenza l'errore in $w_p = w(z_p)$ raggiunge un minimo [5].

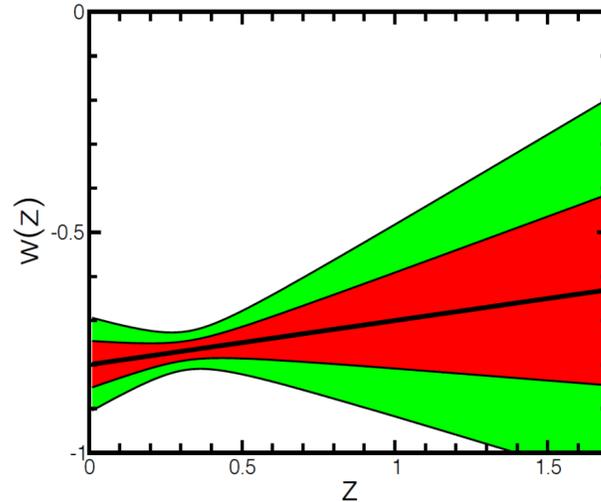


Figura 1.6: Vincoli osservativi su $w(z)$, assumendo $w = w_0 + w'z$. Si noti il valore pivot $z_p \simeq 0.3$ in cui $w(z)$ è meglio determinato [5].

1.6 Tecniche di analisi cosmologiche

Gli sviluppi teorici della cosmologia moderna, partendo dall'equazione fondamentale di campo della Relatività Generale, hanno portato a straordinarie scoperte. Attraverso l'applicazione di questi principi fisici, gli scienziati hanno sviluppato modelli che ci hanno permesso di sondare l'origine, l'evoluzione e la struttura del cosmo. Queste analisi hanno più volte rivoluzionato la nostra visione dell'Universo e continuano ad ispirare nuove domande e ricerche.

Tuttavia, come in ogni campo scientifico, all'aumentare della conoscenza anche la complessità degli studi aumenta. Già la trattazione di un sistema a tre corpi che interagiscono gravitazionalmente non ha una soluzione analitica. Oltre a questo, in cosmologia moderna a causa degli oggetti e dei fenomeni studiati, sono da tenere presente le seguenti condizioni:

- Nessun fenomeno potrà mai essere riprodotto in laboratorio. Tutto lo studio sarà sempre fatto tramite misure indirette.
- Quasi mai le osservazioni sono replicabili. È possibile, certo, osservare più volte lo stesso fenomeno, ma con condizioni a contorno ogni volta differenti.
- In moltissimi casi, basti pensare all'evoluzione di stelle, galassie o ammassi, le osservazioni condotte negli anni riguardano in realtà una situazione sempre pressoché statica, a causa dei tempi di evoluzione studiati che possono avere durate temporali anche di miliardi di anni.
- Le dimensioni spaziali trattate dalla cosmologia sono quelle dell'intero Universo, di conseguenza quello che noi effettivamente siamo in grado di misurare è solo una piccola parte del sistema studiato, cioè l'Universo osservabile.

- Trattandosi di sistemi in condizioni estremamente diverse da quelli con cui abbiamo a che fare nella vita quotidiana - basti pensare alle temperature, alle energie, alle durate temporali e alle dimensioni spaziali -, la fisica che riguarda questi fenomeni non necessariamente coincide con quella che utilizziamo per descrivere i fenomeni a scala terrestre.

Per tutti questi motivi, l'utilizzo dei computer ha svolto un ruolo fondamentale nei calcoli cosmologici, diventando uno strumento non solo comodo, ma ormai necessario, grazie all'utilizzo di software appositi. Tra i vantaggi principali, abbiamo:

- Tempi di calcolo trascurabili rispetto a quelli umani.
- Errori di calcolo trascurabili.
- Possibile risoluzione di problemi privi di soluzione analitica.
- Possibile modellizzazione e simulazione di qualunque sistema fisico. In Figura 1.7 se ne veda un esempio.

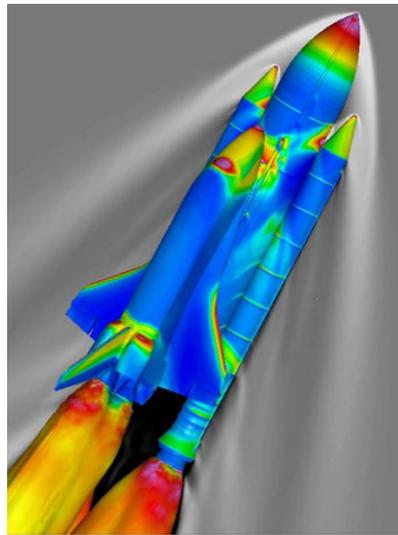


Figura 1.7: Analisi fluidodinamica computazionale dello Space Shuttle, ad un'altezza di 20 000 m. Le superfici del velivolo sono colorate in base al coefficiente di pressione e i contorni grigi rappresentano la variazione di densità dell'aria circostante. I valori sono calcolati con il software OVERFLOW. [16]

Si può quindi dire che oggi l'utilizzo di queste tecnologie ha permesso di fare enormi passi avanti. Proprio per questi motivi, un numero sempre maggiore di ricercatori si dedica proprio allo sviluppo di software.

Anche le librerie numeriche su cui ho svolto il lavoro di Tesi rientra in questo tipo di strumenti di calcolo.

Capitolo 2

Libreria di calcolo CosmoBolognaLib

Le `CosmoBolognaLib` (CBL) sono librerie di calcolo numerico, implementate dal prof. Federico Marulli a partire dal 2010, gestite tramite software `git` - di proprietà del Dipartimento di Fisica ed Astronomia dell'Università di Bologna - e rese pubbliche in `gitlab` dal 2015 ¹ [8]. L'implementazione e validazione delle CBL viene ora portata avanti dal gruppo di ricerca di cosmologia del dipartimento sempre sotto la supervisione del prof. Marulli. Avendo inoltre licenza di tipo *free software*, garantiscono a chiunque la possibilità di implementarle senza vincoli [8]. Le CBL sono dunque in continuo sviluppo, sia dal punto di vista di ottimizzazione degli algoritmi di calcolo, sia dal punto di vista di metodi numerici disponibili.

Le CBL sono scritte in C++, ma forniscono la possibilità di farne uso anche in linguaggio Python. Questo è possibile tramite un *wrapper* implementato in modo automatico tramite il software `SWIG` ². È inoltre possibile utilizzare le librerie dall'interno di un *notebook Colab* offerto da Google, oppure tramite un ambiente virtuale Ubuntu pre-impostato, già contenente le librerie.

Le librerie forniscono inoltre una documentazione che viene generata automaticamente a partire dal codice sorgente del software, tramite il software `doxygen`.

2.1 Struttura e contenuti

Attualmente le CBL contengono numerose classi, includendo algoritmi utili per diversi settori della cosmologia moderna. La complessità del codice implementato richiede una manutenzione complessa. Al fine di semplificare la gestione di queste librerie è stato recentemente deciso di apportare una serie di modifiche, delle quali molte sono ancora in corso. Questo progetto di Tesi ha contribuito in queste attività di ristrutturazione e semplificazione delle CBL, come verrà descritto nel Capitolo 3. Nelle prossime sezioni invece sono descritti gli elementi principali che compongono le CBL.

¹<https://gitlab.com/federicomarulli/CosmoBolognaLib>

²<https://www.swig.org>

2.1.1 Moduli

Le CBL includono una serie di moduli per gestire costanti fisiche e matematiche e altre variabili utili per i metodi numerici implementati. Qui di seguito ne viene fornita una breve, ma completa, descrizione:

- **conversion factors:** modulo composto da circa quaranta elementi, utilizzati per la conversione fra unità di misura o la conversione in `std::strings`. Un esempio è il fattore per convertire da chilometri a metri tramite la variabile:

```
static const double cbl::par::kilo = 1.e3;
```

,

- **mathematical constants:** modulo contenente due costanti matematiche frequentemente utilizzate:

```
static const double cbl::par::pi = 3.1415926535897932;
static const double cbl::par::ee = 2.7182818284590452;
```

- **physical and astronomical constants:** modulo contenente costanti fisiche. Un esempio è la costante di Planck barrata (\hbar):

```
static const double cbl::par::hbar = 1.054571726e-34;
```

- **colours:** modulo contenente un insieme di codici corrispondenti a otto colori, in formato *sequenza di escape ANSI* per gestire messaggi stampati su schermo. Per esempio, per il rosso abbiamo:

```
static const std::string cbl::par::col_red = "\033[0;31m";
```

- **default values:** questo modulo contiene un insieme di variabili che spesso compaiono all'interno del codice. Un esempio è la stringa nulla:

```
static const std::string cbl::par::defaultString = "NULL";
```

- **internal messages:** modulo che contiene i principali messaggi di avvertenza che si utilizzano per segnalare un problema ad una determinata linea di codice. Un esempio è il semplice messaggio di errore:

```
static const std::string errorMsg =
"\n" + col_onred +
" ===== "
+ col_default + "\n" + col_onred +
" ===== \texttt{CosmoBolognaLib} error! ===== "
+ col_default + "\n" + col_onred +
" ===== "
+ col_default + "\n\n";
```

2.1.2 *Namespace*

Un'altra componente fondamentale per ambienti di programmazione con numerose classi e funzioni è quella dei cosiddetti *namespace*. Questi consentono di evitare conflitti tra funzioni e classi con gli stessi nomi ma utilizzati per scopi diversi, ad esempio all'interno di librerie diverse. In questo modo è inoltre più facile riconoscere ogni elemento di interesse nelle CBL, data questa ulteriore suddivisione del codice. Nelle CBL i namespace sono pensati per suddividere le classi presenti in base ai metodi di calcolo implementati.

Il *namespace* delle CBL si chiama `cbl`, definito come "The global namespace of the CosmoBolognaLib"³. Di seguito sono riportati tutti i *namespace* contenuti in esso:

- **catalogue**: namespace contenente le classi che si occupano della gestione dei cataloghi di sorgenti astronomiche.
- **chainmesh**: namespace contenente le classi che si occupano del metodo *chainmesh*, che permette un'efficace organizzazione dei dati di cataloghi per il calcolo di coppie e triplette.
- **cosmology**: namespace contenente le classi utilizzate per il calcolo di funzioni cosmologiche.
- **data**: namespace contenente le classi che si occupano della gestione dei dati. Questi possono essere di diversi tipi e dimensioni; dai semplici dati 1D ai campi scalari 3D, dalle matrici di covarianza ai campi vettoriali.
- **glob**: namespace che contiene tutte le funzioni e le classi di uso generico. Non sono pensate per essere usate dall'utente, ma solo per il lavoro interno di sviluppo ed ottimizzazione.
- **lognormale**: namespace contenente le classi utilizzate per la costruzione di cataloghi *log-normal*.
- **measure**: namespace contenente le classi utilizzate per la gestione di ogni tipo di misura cosmologica.
- **modelling**: namespace contenente le classi utilizzate per le modellizzazioni di dataset cosmologici.
- **pairs**: namespace contenente le classi utilizzate per la gestione ed il controllo di coppie di oggetti, in diversi possibili sistemi.
- **par**: namespace contenente tutti i parametri e le costanti globali, che possono essere utilizzate all'interno di tutta la libreria.
- **random**: namespace contenente le classi utilizzate nella gestione e nel controllo di numeri randomici.

³<https://federicomarulli.github.io/CosmoBolognaLib/Doc/html/index.html>

- **statistics**: namespace contenente le classi utilizzate per l'analisi statistica.
- **triplets**: namespace contenente le classi utilizzate per la gestione di triplette di oggetti, in diversi possibili sistemi.
- **wrapper**: namespace contenente tutto ciò che riguarda il wrapping delle librerie, permettendone l'utilizzo in Python.

2.1.3 Classi

Come anticipato nella sezione precedente, ogni *namespace* contiene al suo interno delle classi. Prima del mio intervento, esisteva un'unica classe `cb1::cosmology::Cosmology` nel *namespace* corrispondente. Come verrà descritto in dettaglio nel Capitolo 3, lo scopo di questo lavoro di Tesi è stato quello di dividere la classe `Cosmology` in varie nuove classi, tutte all'interno del *namespace* `cb1::cosmology`.

2.1.4 File hierarchy

Come già detto precedentemente in questa sezione, i *namespace* e le classi descritte finora hanno lo scopo di dare ordine al codice, semplificando l'utilizzo delle librerie per gli utenti esterni. Inoltre, il codice che compone le librerie non è ovviamente conveniente che sia implementato all'interno di un unico file, essendo composto da decine di migliaia di righe. Per questo motivo, il contenuto delle librerie è stato suddiviso in numerosi file e cartelle, seguendo un preciso ordine. Di seguito è descritta l'impostazione delle directory di questa libreria.

La struttura basilare è abbastanza semplice. All'interno della cartella *CosmoBolognaLib*, che contiene tutte le librerie, è presente una cartella di nome *Headers* che contiene tutti i file header (.h), contenenti tutte le dichiarazioni delle classi delle CBL. Al suo stesso livello, sono presenti tutte le cartelle, con i nomi coincidenti al nome dei file .h, contenenti i rispettivi file .cpp, che ne contengono le implementazioni. Ad aggiungersi a queste, troviamo poi la cartella *Doc*, che contiene file di documentazione. È presente anche la cartella *Examples*, contenente altrettante cartelle, contenenti esempi e dimostrazioni sull'utilizzo del codice che compone l'intera libreria. Un'altra cartella presente è denominata *Python* e contiene file .i necessari per la conversione a questo linguaggio di programmazione. Oltre le cartelle, sono presenti pochi altri file: quelli necessari per il settaggio dell'ambiente di programmazione e compilazione, il *Makefile* e *test.py*. Il *Makefile* è quello responsabile della compilazione delle librerie, in diverse possibili modalità (nel seguente capitolo vedremo meglio quali); il *test.py* ha invece lo scopo di verificare il funzionamento delle CBL. Eseguendo quest'ultimo file infatti, inizia un processo di compilazione per poi verificarne la funzionalità tramite tutti gli esempi presenti, la corretta versione in Python ed evidenziando molte potenziali fonti di problemi.

2.2 Esempio di utilizzo

In questa sezione, andremo ad analizzare un esempio che spiega come utilizzare le CBL. Naturalmente, non è possibile comprendere tutte le funzionalità delle CBL all'interno di un programma di qualche decina di righe, ma le modalità di utilizzo sono comunque le stesse per ogni classe, e dunque questo esempio risulta piuttosto generale.

Nel codice seguente, in linguaggio C++, sarà trattata la classe *Cosmology*, che è proprio quella su cui ho lavorato in questo progetto di Tesi. Andremo a considerare il calcolo del bias di aloni di materia oscura.

Innanzitutto, è necessario importare la classe della libreria di nostro interesse e, come in ogni programma C++, aprire il *main* in cui andremo a scrivere il codice da eseguire. Nel nostro caso, la classe è denominata *Bias*. Il codice inizierà quindi con:

```
#include "Bias.h"

int main () {

    return 0;

}
```

A questo punto, è necessario definire il modello cosmologico in cui lavorare. Per fare questo, ci sono diversi costruttori possibili; qui utilizzeremo quello in cui è sufficiente fornire un modello cosmologico di default. Assumeremo il modello Λ CDM con parametri cosmologici di Planck denominato *_Planck18_* (attualmente sono disponibili numerosi modelli cosmologici di default ⁴). Il codice per creare l'oggetto di classe *Cosmology* con cui fare tutto questo è il seguente:

```
cbl::cosmology::Cosmology cosmology
    {cbl::cosmology::CosmologicalModel::_Planck18_};
```

Da notare come la classe *Cosmology* sia all'interno del namespace *cosmology*, che è a sua volta all'interno del namespace *cbl*, come già introdotto nella sezione precedente.

Fatto questo, è necessario istanziare un oggetto di classe *Bias*, contenente la funzione che vogliamo utilizzare. Procediamo quindi aggiungendo la riga:

```
cbl::cosmology::Bias bias(cosmology);
```

Per utilizzare la funzione che misura il bias degli aloni di materia oscura, denominata *bias_halo*, è sufficiente implementare la riga:

```
const double bias_halo =
    bias.bias_halo(1.e13, 1., "ST99", "EisensteinHu");
```

in cui andiamo a creare un valore *double* che sarà proprio il nostro dato cercato. La funzione in questione richiede quattro valori in input: massa, redshift, modello di bias e Boltzmann solver; forniti questi, la funzione può svolgere i calcoli, assegnando il risultato alla variabile *bias_halo*. In realtà la funzione può ricevere anche altri dati in input, che se non assegnati assumono valori di default.

⁴<https://federicomarulli.github.io/CosmoBolognaLib/Doc/html/index.html>

Ora, aggiungendo solo la scrittura a terminale del risultato ottenuto, il programma completo risulta essere:

```
#include "Bias.h"

int main () {

    cbl::cosmology::Cosmology cosmology
        {cbl::cosmology::CosmologicalModel::_Planck18_};

    cbl::cosmology::Bias bias(cosmology);

    const double bias_halo =
        bias.bias_halo(1.e13, 1., "ST99", "EisensteinHu");

    std::cout <<
        "The halo bias at M=1e13 Msun/h and z=1,
        predicted by the Sheth & Tormen (1999) model,
        is:" << bias_halo << std::endl;

    return 0;
}
```

Questo è un programma molto semplice, che tramite l'utilizzo di una sola funzione ottiene il risultato desiderato. Tuttavia, è sufficientemente generale per evidenziare la struttura di un qualsiasi codice che usi le CBL.

Capitolo 3

Ristrutturazione della classe Cosmology

Lo scopo del mio lavoro di Tesi è stato quello di ottimizzare la struttura ed il funzionamento della classe *Cosmology* delle CBL, descritte nel Capitolo 2.

In questo capitolo, descriveremo qual era la struttura di questa classe prima del mio intervento, evidenziandone gli svantaggi principali; andremo poi ad analizzare, passo per passo, le modifiche apportate, descrivendo il metodo di lavoro utilizzato; infine, confronteremo il funzionamento della classe prima e dopo l'aggiornamento.

3.1 Situazione iniziale

Inizialmente, la classe *Cosmology* aveva un totale di 352 elementi al suo interno, di cui 326 erano funzioni membro; queste erano definite in 16 files *.cpp*, in base ai metodi numerici implementati. Tra queste funzioni, erano presenti ¹:

- 2 costruttori
- 1 distruttore
- 32 funzioni di `get` per ottenere i membri privati della classe
- 25 funzioni di `set` per settare i membri privati della classe
- 48 funzioni di calcolo di parametri cosmologici
- 15 funzioni di calcolo di distanze e volumi cosmologici
- 17 funzioni di calcolo di statistiche di aloni di materia oscura e altre quantità relative
- 7 funzioni di calcolo di accrescimento di massa su aloni di materia oscura
- 21 funzioni di calcolo dello spettro di potenza e altre quantità relative

¹<https://federicomarulli.github.io/CosmoBolognaLib/Doc/html/index.html>

- 2 funzioni di calcolo del profilo di densità di aloni di materia oscura
- 26 funzioni di calcolo di funzioni di correlazione, bias ed altre quantità relative a statistiche a due punti
- 24 funzioni di modellizzazione delle distorsioni redshift-space
- 6 funzioni di modellizzazione delle oscillazioni acustiche barioniche
- 9 funzioni di modellizzazione di quantità cosmologiche in cosmologie con non-gaussianità primordiale
- 7 funzioni di calcolo della *void size function* dei vuoti cosmici
- 3 funzioni di calcolo di matrici di covarianza per multipoli
- 22 funzioni di calcolo di spettro di potenza non lineare
- 41 funzioni di calcolo di funzioni di correlazioni a tre punti
- 18 funzioni ausiliarie di utilizzo interno (nella sezione privata)

Appare evidente come, all'interno di un'unica classe, fossero presenti numerose funzioni legate ad utilizzi e scopi diversi, e non tutte localizzate sul calcolo numerico di quantità cosmologiche. Il problema non era tanto nell'utilizzo della classe da parte dell'utente, il cui unico svantaggio era il tempo necessario per trovare una determinata funzione, sulle 326 disponibili. Il più grosso svantaggio era invece a livello di mantenibilità ed estendibilità del codice. Essendoci dieci diverse categorie di metodi numerici implementati, la soluzione più vantaggiosa è risultata quella di costruire nuove classi specifiche per ognuna di queste, così da alleggerire la classe *Cosmology*. In questo modo, è possibile includere nel codice solo la classe contenente le funzioni che implementano gli algoritmi su cui si vuole lavorare, semplificando la struttura dei codici e facilitandone futuri sviluppi.

3.2 Strategia utilizzata

Dopo alcune prove in ambienti locali, scollegati cioè dal *repository gitdifa*, ho stilato una lista di operazioni da eseguire per implementare una nuova classe, separata da *Cosmology*. Le modifiche principali sono le seguenti:

- **Costruzione del nuovo header:** per prima cosa è necessario creare il file *.h* che conterrà la struttura della nuova classe e le dichiarazioni dei suoi membri.
- **Modifica del *.cpp* relativo:** successivamente, è necessario creare il file contenente le implementazioni delle funzioni membro, dichiarate al punto precedente.

- **Modifiche di tutte le altre classi coinvolte:** inizia qui una delle parti più delicate del lavoro, perché le funzioni di ogni classe spesso sono utilizzate in altre classi. Per questo motivo, è necessario correggere tutte le parti di codice in cui ogni funzione membro della classe appena creata viene utilizzata.
- **CBL.h:** La nuova classe va inclusa nel file `CBL.h` in modo da poter essere importata insieme a tutte le altre delle CBL.
- **Conversione in Python:** una volta verificato che il funzionamento delle librerie CBL in formato C++ sia corretto, è necessario creare un file `.i`, o modificarne uno già esistente, nella cartella *Python*, per permettere la conversione automatica in Python della nuova classe tramite `swig`.
- **Scrittura di nuovi esempi C++ e Python:** è necessario ora scrivere piccoli programmi di esempio per verificare la funzionalità della nuova classe appena creata e per mostrarne l'utilizzo ad un utente esterno. Il tutto sia in linguaggio C++ che in linguaggio Python, anche per verificare la corrispondenza dei risultati ottenuti.
- **Correzioni finali:** spesso, a causa delle modifiche fatte, possono comparire warning ed errori che, senza un apposito controllo, verrebbero ignorati in fase di compilazione. Per questo motivo, è stato creato uno script apposito per scrivere su un file tutti i possibili warning e le potenziali anomalie presenti nel codice (mi riferisco a `test.py`, descritto nel Capitolo 2). Grazie a questo script, è possibile verificare che la nuova classe sia stata correttamente implementata all'interno delle librie.

Nelle seguenti sezioni, verranno mostrati esempi che spiegano in cosa consiste ognuna di queste operazioni. Per ragioni di spazio e comprensione, non sarà riportato l'intero codice scritto e/o modificato, ma solo le righe di codice più significative.

3.2.1 Costruzione del nuovo header

La struttura classica di ogni nuovo header quasi sempre la seguente:

```
#ifndef __NAMECLASS__
#define __NAMECLASS__

#include "Cosmology.h"

namespace cbl {

    namespace cosmology {

        class NameClass {

        private:
```

```

// object of class Cosmology
cbl::cosmology::Cosmology m_cosmology;

// shared pointer to an object of class Cosmology
std::shared_ptr<cosmology::Cosmology> m_cosmology = NULL;

public:

    NameClass () = default;

    NameClass (cbl::cosmology::Cosmology cosmology);

    ~NameClass () = default;

    // statements of other functions

};
}
}

```

Questa nuova classe conterrà, come segnato nell'ultimo commento del codice appena mostrato, tutte le dichiarazioni delle funzioni implementate nel file `.cpp` corrispondente. Questa suddivisione delle definizioni delle funzioni era già presente; le funzioni erano cioè già separate in file diversi proprio secondo la divisione che anche le classi future avrebbero dovuto avere.

Il codice appena mostrato è stato in realtà minimizzato per chiarezza; in quello vero sono presenti commenti esplicativi sul funzionamento di ogni singolo membro presente e riguardanti informazioni generali delle librerie. È importante sottolineare che in tutte le nuove classi è stato necessario includere la classe *Cosmology* e aggiungere un oggetto di quel tipo come membro del costruttore o, in alcuni casi, utilizzare un puntatore ad un oggetto di quel tipo come riferimento. Questo è dovuto al fatto che tutte queste nuove classi, per quanto relative a metodi diversi, fanno sempre riferimento ad un particolare modello cosmologico, su cui eseguono i calcoli. In molti casi, tuttavia, è stato necessario includere anche altri file, a causa dell'utilizzo di altre funzioni per eseguire i calcoli interni alla classe. Un esempio può essere il caso di *HaloProfile.h*, che esegue alcuni calcoli numerici tramite l'utilizzo di un modello di bias degli aloni; per questo motivo, è stato necessario includere in quel caso anche *Bias.h*.

Faccio infine notare come siano sempre presenti anche nelle nuove classi i *namespace* per gli stessi motivi.

3.2.2 Modifica del `.cpp` relativo

Il file *NameClass.cpp*, relativo alla classe *NameClass* descritta nella precedente sezione ed implementata nell'apposito header *NameClass.h*, richiede le seguenti modifiche:

- **Inclusione del `.h` corretto:** è necessario sostituire la linea di codice:

```
#include "Cosmology.h"
```

con la seguente:

```
#include "NameClass.h"
```

NameClass.h è infatti il nuovo file che contiene le definizioni delle funzioni implementate all'interno del file *NameClass.cpp*.

- **Correzione dei riferimenti alle funzioni:** è necessario sostituire il *namespace* modificando le seguenti righe di codice:

```
type cbl::cosmology::Cosmology::name_function (members)
{
    ...
}
```

con:

```
type cbl::cosmology::NameClass::name_function (members)
{
    ...
}
```

- **Richiamo di elementi di *Cosmology*:** come già accennato, all'interno di queste nuove classi si usano membri e parametri di *Cosmology*. Essendo in precedenza questo codice all'interno della classe stessa, naturalmente era sufficiente utilizzare il nome dei parametri, anche se di tipo *private*. Ora, siamo invece in una nuova classe; per questo motivo, se prima era sufficiente utilizzare il nome della variabile, ad esempio:

```
m_Omega_neutrinos
```

per riferirsi alla densità di neutrini in una data cosmologia (che è una variabile privata), ora è necessario utilizzare la relativa funzione di *get* tramite il comando:

```
m_cosmology.Omega_neutrinos()
```

in cui tramite l'oggetto *m_cosmology*, possiamo accedere alla sua funzione pubblica *Omega_neutrinos()*, implementata specificatamente per poter accedere a quel dato.

3.2.3 Modifiche delle altre classi presenti

Dopo aver creato il nuovo file header (*.h*) e aver modificato di conseguenza il file *.cpp*, come descritto nelle sezioni precedenti, è necessario adattare tutte le altre classi della CBL che utilizzano membri di quella appena creata. Prendiamo in esempio la classe *MassFunction*, che in alcune sue funzioni considera cosmologie con non-gaussianità primordiale. A causa di questo, *MassFunction* richiede l'inclusione dell'header *NG.h* (che contiene la classe *NG* dedicata) e ne utilizza diverse funzioni. Di conseguenza, dopo la

creazione della classe *NG*, sono state necessarie modifiche all'interno di *MassFunction.h* e *MassFunction.cpp* per renderne possibile l'utilizzo. Il concetto base è lo stesso di quello trattato al punto precedente, l'unica differenza è che invece di sostituire elementi della classe stessa con elementi di una classe esterna, qui è necessario sostituire elementi di una classe esterna con elementi di un'altra classe esterna differente. Il codice, quindi, passerà ad esempio da:

```
cosmology.MF_correction(members);
```

a:

```
NG ng(m_cosmology);
ng.MF_correction(members);
```

Dunque invece di utilizzare la funzione dentro la classe *Cosmology*, è necessario istanziare un oggetto *NG* e utilizzare una sua funzione.

3.2.4 Aggiunta a *CBL.h* e altri file secondari

Il nuovo header file (*.h*) contenente la nuova classe creata va aggiunto nel file *CBL.h* che contiene tutti gli header delle CBL. Inoltre è necessario modificare i file di script *em* e *vm*. Questi consentono di aprire tutti i file di una classe (cioè *.h* e *.cpp*) con i codici *emacs* e *gvim*. È di conseguenza necessario, dopo la creazione di una classe, inserire i nuovi file all'interno di questi script.

3.2.5 Conversione in Python

È infine arrivato il momento di convertire in Python questa nuova classe appena creata. Esiste una cartella apposita denominata *Python* con la sottocartella *Lib*, che contiene tutti i file *.i* necessari per rendere possibile la creazione automatica di un wrapper tramite *swig*. Questi file hanno un contenuto abbastanza semplice: all'inizio è riportato il nome del modulo trattato, seguito dall'*import* dei file necessari per far funzionare il codice ed infine l'inclusione dei file su cui si sta lavorando. Per comodità, ho scelto di manetene tutte le classi appena create all'interno del modulo *cblCosmology*, il cui file *Cosmology.i* era già presente. In questo, è stato sufficiente aggiungere il nome del *.h* appena creato nell'elenco presente:

```
%module cblCosmology

%include "Path.i"
%import "Kernel.i"
%import "Data.i"

%shared_ptr(cbl::cosmology::Cosmology);

%{
#include "Data.h"
...
}
```

```

...
...
#include "NameClass.h"

%}

#include "Data.h"
...
...
...
#include "NameClass.h"

```

In questo modo anche il nuovo file *NameClass.h* sarà compreso nel wrapping.

3.2.6 Scrittura di nuovi esempi C++ e Python

Per verificare il corretto funzionamento della nuova classe appena creata e mostrare come utilizzarla è ora conveniente creare uno o più esempi (sia in C++ che in Python).

I due elementi che devono sempre essere presenti in un esempio perché questo sia utile ed esplicativo sono:

- la costruzione di un oggetto della nuova classe
- l'utilizzo di una sua funzione membro

Trattiamo ora l'esempio che ho scritto per la classe *NG*. Questa classe ha lo scopo di trattare il calcolo di cosmologie con non-gaussianità primordiale (da cui l'abbreviazione *NG*). Assumiamo in particolare di voler calcolare il valore della skewness del campo di densità di materia in cosmologia, con non-gaussianità primordiale.

Dopo l'inclusione della classe necessaria ed apertura del main tramite:

```
#include "NG.h"
```

```
int main () {
```

costruiamo innanzitutto l'oggetto di classe *Cosmology* con:

```
cbl::cosmology::Cosmology cosmology
    {cbl::cosmology::CosmologicalModel::_Planck18_};
```

per poi costruirne l'oggetto della nuova classe *NG* con:

```
cbl::cosmology::NG ng(cosmology);
```

A questo punto, è sufficiente richiamare la relativa funzione membro *skewness*, che calcola la quantità desiderata. Per il suo utilizzo è necessario fornire in input la massa degli aloni di materia oscura ed il metodo di calcolo desiderato per stimare lo spettro di potenza lineare della materia. Assumiamo ad esempio una massa pari a $10^{13}M_{\odot}/h$ ed il Boltzmann solver *EisensteinHu*. Con questi dati il codice risulta:

```
const double SK = ng.skewness(1.e13, "EisensteinHu");
```

Ho definito questo valore come costante perché all'interno di questo esempio non dovrà variare, ma solo essere stampato a riga di comando.

Possiamo quindi andare a stampare questo valore a schermo con il comando:

```
std::cout << "The skewness at M=1e13 Msun/h is:"
<< SK << std::endl << std::endl;
```

e successivamente chiudere il main:

```
return 0;
}
```

Questo codice di esempio può infine essere incluso all'interno della sezione *try-catch*; in questo modo, in caso di errore nell'esecuzione, viene restituito il messaggio fornito nella sezione *catch* e di conseguenza lo sviluppatore può capire quale classe delle CBL è responsabile dell'errore.

Analogamente è possibile scrivere lo stesso programma di esempio anche in linguaggio *Python*, così da verificare anche la compatibilità dei risultati. Allo stesso modo di prima, iniziamo con l'importare la libreria:

```
import \texttt{CosmoBolognaLib} as cbl
```

Andiamo poi a definire la cosmologia con

```
Planck18 = cbl.Cosmology(cbl.CosmologicalModel_Planck18_)
```

e l'oggetto per gestire calcoli in cosmologie con non-gaussianità primordiali

```
ng = cbl.NG(Planck18)
```

A questo punto, assumendo gli stessi valori assunti nel formato precedente - così da verificare poi la corrispondenza dei risultati - il codice per ottenerla quantità voluta risulta essere:

```
SK = ng.skewness(1.e13, "EisensteinHu")
```

e da qui si può procedere direttamente con la stampa tramite:

```
print("The skewness at M=1e13 Msun/h is:", SK)
```

In assenza di errori ci aspettiamo che i due valori ottenuti con gli esempi in C++ e Python siano uguali, a meno di differenze nel numero di cifre significative utilizzate.

3.2.7 Controlli finali

È ora il momento di verificare che tutte le librerie CBL funzionino correttamente, nonostante le modifiche apportate. Solitamente sono necessarie numerose compilazioni in corso di modifica per riconoscere velocemente possibili malfunzionamenti.

Innanzitutto, per far sì che i nuovi esempi appena inseriti siano inclusi nella documentazione, è necessario aggiungere alcune linee di codice all'interno di *Kernel.h*:

```
/**
 * @example name_example.cpp
 *

```

```
* Description of the example
*/
```

Inoltre è necessario aggiungere nel Makefile delle CBL il comando per la compilazione dei nuovi esempi. Nel caso dell'esempio presente nel file `fsigma8.cpp`:

```
$(call colorecho, "\n" Compiling the example code: fsigma8.cpp
... "\n")
cd $(PWD)/Examples/cosmology ; make fsigma8 CXX=$(CXX)
FLAGS_INC='$(FLAGS_INC)'
```

A questo punto, si può verificare il completo funzionamento delle CBL tramite lo script `test.py`, dopo aver incluso i nuovi esempi al suo interno, con il comando:

```
./test.py ALL
```

In questo modo, lo script `test.py` eseguirà le seguenti azioni:

- **make**: compilazione *C++*
- **make python**: wrapping *Python*
- **make allExamples**: compilazione di tutti gli esempi presenti
- **make documentation**: scrittura della documentazione su vari file `html`, contenente anche tutti i potenziali *warning* che potrebbero uscire in un futuro utilizzo delle librerie
- Esecuzione di tutti gli esempi presenti per verificarne il corretto funzionamento

Se questo comando termina con successo e non sono presenti warning all'interno del file `.../Doc/WARNING`, la libreria si può considerare perfettamente funzionante.

3.3 Nuove classi per le CosmoBolognaLib

Nella Sezione 3.2 abbiamo descritto tutte le operazioni necessarie per la creazione di una nuova classe all'interno delle librerie CBL. Seguendo questi passaggi, ho costruito otto nuove classi:

- 3PCF (41 funzioni)
- BAO (9 funzioni)
- BIAS (17 funzioni)
- MassFunction (21 funzioni)
- MassGrowth (7 funzioni)
- NG (9 funzioni)

- RSD (14 funzioni)
- SizeFunction (11 funzioni)
- SuperSampleCovariance (10 funzioni)
- Velocities (4 funzioni)

Nel complesso, oltre due terzi delle funzioni membro della classe *Cosmology* sono ora diventate delle nuove classi, create in questo lavoro di Tesi.

In alcuni casi particolari sono stati riscontrati errori causati da vecchie classi attualmente inutilizzate o prossime all'eliminazione. In questi casi, si è deciso di non correggere gli errori in quanto non necessario. Un esempio è la classe *CosmClassFunc*, contenente algoritmi di calcolo numerico già sostituiti in quasi tutte le librerie presenti.

Questo lavoro rappresenta il primo passo di una più generale ristrutturazione della classe *Cosmology*.

Capitolo 4

Conclusioni

Come descritto nella Sezione 3.1, quando è iniziato questo progetto di Tesi all'interno delle CBL era presente un'unica classe *Cosmology*, contenente tutti i modelli cosmologici e relative funzioni implementate nelle librerie. Lo scopo di questo lavoro di Tesi è consistito nella ristrutturazione di questa classe, tramite la creazione di nuove classi per ogni insieme di funzioni presenti (nella sezione precedente - Sezione 3.3 - è presente un elenco delle nuove classi disponibili oltre a *Cosmology*). Queste hanno tutte un membro privato di classe *Cosmology* per gestire i calcoli cosmologici, ma al loro interno sono presenti solo le funzioni riguardanti uno specifico insieme di algoritmi di calcolo. In questo modo sarà possibile effettuare implementazioni future al codice in maniera più semplice e veloce.

Quando sarà terminata la creazione di queste nuove classi, è in programma la ristrutturazione della stessa classe *Cosmology*, che diventerà una classe interfaccia per gestire classi figlie che implementeranno diversi modelli cosmologici. Queste saranno uno strumento fondamentale nella struttura del codice, offrendo diversi vantaggi:

- Ereditarietà: le sotto-classi potranno ereditare le caratteristiche ed i comportamenti delle classi di base.
- Riutilizzo del codice: con l'ereditarietà si potrà utilizzare il codice già esistente nella classe di base.
- Polimorfismo: un oggetto della sotto-classe potrà essere usato direttamente come oggetto della classe madre.
- Estensione: con questo formato di sotto-classi, ogni modello cosmologico aggiungerà alla classe *Cosmology* i propri membri, ma solo lungo il suo ramo.

Con questa nuova struttura, la gerarchia del codice sarà molto più chiara e definita, e, soprattutto, lo sviluppo futuro del codice risulterà molto più semplice e veloce; sarà infatti possibile aggiungere nuove funzionalità alle sotto-classi senza dover modificare il codice esistente nella classe di base, riducendo così il rischio di introdurre errori o problemi imprevisti.

Bibliografia

- [1] Håvard Alnes, Morad Amarzguioui, and Øyvind Grøn. Inhomogeneous alternative to dark energy? *Physical Review Letters*, 73(8):083519, April 2006.
- [2] Gianfranco Bertone. *Particle Dark Matter*. 2013.
- [3] A. Carati and L. Galgani. Faraway matter as a possible substitute for dark matter. *Workshop Series of the Asociacion Argentina de Astronomia*, 3:277–287, January 2011.
- [4] G. Dvali, G. Gabadadze, and M. Porrati. 4D gravity on a brane in 5D Minkowski space. *Physics Letters B*, 485(1-3):208–214, July 2000.
- [5] J. A. Frieman, M. S. Turner, and D. Huterer. Dark energy and the accelerating universe. *Annual Review of Astronomy and Astrophysics*, 46:385–432, September 2008.
- [6] Edward W. Kolb, Sabino Matarrese, and Antonio Riotto. On cosmic acceleration without dark energy. *New Journal of Physics*, 8(12):322, December 2006.
- [7] Eric V. Linder. Exploring the Expansion History of the Universe. *Physical Review Letters*, 90(9):091301, March 2003.
- [8] F. Marulli, A. Veropalumbo, and M. Moresco. CosmoBolognaLib: C++ libraries for cosmological calculations. *Astronomy and Computing*, 14:35–42, January 2016.
- [9] Planck Collaboration, N. Aghanim, Y. Akrami, M. Ashdown, J. Aumont, C. Baccigalupi, M. Ballardini, A. J. Banday, R. B. Barreiro, N. Bartolo, S. Basak, R. Battye, K. Benabed, J. P. Bernard, M. Bersanelli, P. Bielewicz, J. J. Bock, J. R. Bond, J. Borrill, F. R. Bouchet, F. Boulanger, M. Bucher, C. Burigana, R. C. Butler, E. Calabrese, J. F. Cardoso, J. Carron, A. Challinor, H. C. Chiang, J. Chluba, L. P. L. Colombo, C. Combet, D. Contreras, B. P. Crill, F. Cuttaia, P. de Bernardis, G. de Zotti, J. Delabrouille, J. M. Delouis, E. Di Valentino, J. M. Diego, O. Doré, M. Douspis, A. Ducout, X. Dupac, S. Dusini, G. Efstathiou, F. Elsner, T. A. Enßlin, H. K. Eriksen, Y. Fantaye, M. Farhang, J. Fergusson, R. Fernandez-Cobos, F. Finelli, F. Forastieri, M. Frailis, A. A. Fraisse, E. Franceschi, A. Frolov, S. Galeotta, S. Galli, K. Ganga, R. T. Génova-Santos, M. Gerbino, T. Ghosh, J. González-Nuevo, K. M. Górski, S. Gratton, A. Gruppuso, J. E. Gudmundsson, J. Hamann, W. Handley, F. K. Hansen, D. Herranz, S. R. Hildebrandt, E. Hivon,

- Z. Huang, A. H. Jaffe, W. C. Jones, A. Karakci, E. Keihänen, R. Keskitalo, K. Kiiveri, J. Kim, T. S. Kisner, L. Knox, N. Krachmalnicoff, M. Kunz, H. Kurki-Suonio, G. Lagache, J. M. Lamarre, A. Lasenby, M. Lattanzi, C. R. Lawrence, M. Le Jeune, P. Lemos, J. Lesgourgues, F. Levrier, A. Lewis, M. Liguori, P. B. Lilje, M. Liley, V. Lindholm, M. López-Cañiego, P. M. Lubin, Y. Z. Ma, J. F. Macías-Pérez, G. Maggio, D. Maino, N. Mandolesi, A. Mangilli, A. Marcos-Caballero, M. Maris, P. G. Martin, M. Martinelli, E. Martínez-González, S. Matarrese, N. Mauri, J. D. McEwen, P. R. Meinhold, A. Melchiorri, A. Mennella, M. Migliaccio, M. Millea, S. Mitra, M. A. Miville-Deschênes, D. Molinari, L. Montier, G. Morgante, A. Moss, P. Natoli, H. U. Nørgaard-Nielsen, L. Pagano, D. Paoletti, B. Partridge, G. Patanchon, H. V. Peiris, F. Perrotta, V. Pettorino, F. Piacentini, L. Polastri, G. Polenta, J. L. Puget, J. P. Rachen, M. Reinecke, M. Remazeilles, A. Renzi, G. Rocha, C. Rosset, G. Roudier, J. A. Rubiño-Martín, B. Ruiz-Granados, L. Salvati, M. Sandri, M. Savelainen, D. Scott, E. P. S. Shellard, C. Sirignano, G. Sirri, L. D. Spencer, R. Sunyaev, A. S. Suur-Uski, J. A. Tauber, D. Tavagnacco, M. Tenti, L. Toffolatti, M. Tomasi, T. Trombetti, L. Valenziano, J. Valiviita, B. Van Tent, L. Vibert, P. Vielva, F. Villa, N. Vittorio, B. D. Wandelt, I. K. Wehus, M. White, S. D. M. White, A. Zacchei, and A. Zonca. Planck 2018 results. VI. Cosmological parameters (Corrigendum). *Astronomy and Astrophysics*, 652:C4, August 2021.
- [10] Casadio R. Elements of general relativity, 2021.
- [11] Adam G. Riess, Alexei V. Filippenko, Peter Challis, Alejandro Clocchiatti, Alan Diercks, Peter M. Garnavich, Ron L. Gilliland, Craig J. Hogan, Saurabh Jha, Robert P. Kirshner, B. Leibundgut, M. M. Phillips, David Reiss, Brian P. Schmidt, Robert A. Schommer, R. Chris Smith, J. Spyromilio, Christopher Stubbs, Nicholas B. Suntzeff, and John Tonry. Observational Evidence from Supernovae for an Accelerating Universe and a Cosmological Constant. *Astronomical Journal*, 116(3):1009–1038, September 1998.
- [12] Barbara Ryden. *Introduction to Cosmology*. 2016.
- [13] NASA / WMAP Science Team.
- [14] Robert M. Wald. *General Relativity*. 1984.
- [15] Wikipedia. Espansione metrica dello spazio.
- [16] Wikipedia. Space shuttle.
- [17] Wikipedia. Storia dell’universo.
- [18] F. Zwicky. Die Rotverschiebung von extragalaktischen Nebeln. *Helvetica Physica Acta*, 6:110–127, January 1933.