Propagation de la lumière dans un Univers structuré et nouvelles approches numériques en cosmologie

Vincent Reverdy

▶ **To cite this version:**

Vincent Reverdy. Propagation de la lumière dans un Univers structuré et nouvelles approches numériques en cosmologie. Astrophysique [astro-ph]. Observatoire de Paris, 2014. Français. ⟨NNT : ⟩. ⟨tel-02095297⟩

HAL Id: tel-02095297

https://hal.science/tel-02095297v1

Submitted on 10 Apr 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HAL Authorization

Propagation de la lumière dans un Univers structuré et nouvelles approches numériques en cosmologie

Vincent Reverdy

sous la direction de Jean-Michel Alimi et Yann Rasera

OBSERVATOIRE DE PARIS
9179
BIBLIOTHEQUE

Table des matières

***

Avant-propos i

1 L'Univers homogène : modèle de concordance et extensions 1
- 1.1 Élements de relativité générale et métrique FLRW . . . 2
  - 1.1.1 Équation de champ d'Einstein . . . 2
  - 1.1.2 Équations de Friedmann-Lemaître . . . 3
  - 1.1.3 Contenu de l'Univers . . . 4
  - 1.1.4 Quantités cosmologiques . . . 6
- 1.2 Modèle de concordance ΛCDM . . . 8
  - 1.2.1 Élaboration du modèle . . . 8
  - 1.2.2 Matière noire . . . 9
  - 1.2.3 Énergie noire . . . 11
  - 1.2.4 Propriétés dynamiques . . . 13
- 1.3 Au-delà de la constante cosmologique . . . 16
  - 1.3.1 Validité du modèle de concordance et alternatives . . . 16
  - 1.3.2 Quintessence et autres équations d'état . . . 18
  - 1.3.3 Développements autour de la gravité de Brans-Dicke . . . 19

2 L'Univers structuré : effondrement gravitationnel et sondes cosmologiques 25
- 2.1 Formation des grandes structures . . . 26
  - 2.1.1 Grands relevés observationnels et Univers local . . . 26
  - 2.1.2 Fluctuations primordiales de densité et régime linéaire . . . 28
  - 2.1.3 Régime non-linéaire et scénario hiérarchique de formation . . . 30
- 2.2 Simuler l'évolution des grandes structures . . . 31
  - 2.2.1 Fondements et principes des simulations cosmologiques . . . 31
  - 2.2.2 Caractérisation des champs et des halos de matière noire . . . 33
  - 2.2.3 Note sur le rôle des baryons . . . 36
- 2.3 Sonder la cosmologie au moyen de la structuration . . . 38
  - 2.3.1 Empreinte de la cosmologie sur la formation des structures avec une statistique maximale . . . 38
  - 2.3.2 Propriétés du champ de matière : l'exemple du spectre de puissance . . . 38
  - 2.3.3 Les oscillations acoustiques des baryons : un échelle des distances standard ? . . . 39
  - 2.3.4 Propriétés des halos, l'exemple de la fonction de masse . . . 41
  - 2.3.5 De la nécessité de nouvelles sondes dans l'espace des redshifts . . . 42

3 L'Univers perturbé : de la métrique en champ faible aux équations géodésiques 45
- 3.1 Contexte et problématique : des inhomogénéités au lentillage gravitationnel . . . 46
  - 3.1.1 Inhomogénéités et cosmologie observationnelle . . . 46
  - 3.1.2 Approximations usuelles . . . 51

3.1.3 Problématique . . . 53
3.2 Perturbations cosmologiques engendrées par la structuration . . . 54
3.2.1 Perturbation de la métrique et connexions affines . . . 54
3.2.2 Tenseur d'Einstein et tenseur énergie-impulsion . . . 56
3.2.3 Équations d'Einstein perturbées . . . 58
3.2.4 Quantités calculées dans les simulations cosmologiques . . . 61
3.3 Établissement des équations géodésiques sous-tendant la propagation de la lumière . . . 61
3.3.1 Cas général . . . 61
3.3.2 Cas des géodésiques nulles . . . 64
3.3.3 Formulation alternative . . . 64
3.3.4 Expression du redshift . . . 65
3.4 Implications cosmologiques : formulations analytiques et interprétations . . . 66
3.4.1 Correspondance entre distributions de matière simulée et observée . . . 66
3.4.2 Effets sur le redshift et sur la distance de luminosité . . . 66

4 L'Univers observé : empreinte des inhomogénéités sur les mesures de distance 69
4.1 Méthodologie et approximations . . . 70
4.1.1 Principe du backward raytracing en champ faible . . . 70
4.1.2 Obtention des jeux de données . . . 74
4.1.3 Hypothèses et approximations . . . 76
4.2 Résultats . . . 80
4.2.1 Effets sur le redshift . . . 80
4.2.2 Effets sur les distances cosmologiques . . . 81
4.3 Implications cosmologiques et perspectives . . . 86
4.3.1 Bilan . . . 86
4.3.2 Perspectives . . . 86
4.3.3 Conclusion . . . 87

5 Vers de nouvelles architectures logicielles pour la cosmologie numérique 89
5.1 Cosmologie numérique et simulations cosmologiques . . . 90
5.1.1 Contexte et caractéristiques des simulations . . . 90
5.1.2 Déroulement des simulations . . . 93
5.2 Le projet Dark Energy Universe Simulation . . . 94
5.2.1 Problématiques et objectifs . . . 94
5.2.2 Codes et procédures . . . 94
5.3 Le Full Universe Run : une expérience numérique . . . 95
5.3.1 Configuration . . . 95
5.3.2 Temps de calcul et communications . . . 98
5.3.3 Mémoire . . . 98
5.3.4 Entrées et sorties . . . 101
5.3.5 Pré et post-traitements . . . 103
5.4 Limitations de la cosmologie numérique . . . 103
5.4.1 Des succès aux limitations . . . 103
5.4.2 Leçons du Full Universe Run . . . 104
5.4.3 À la source de tous les maux : l'intrication . . . 107
5.4.4 Problématique . . . 108
5.5 Programmation générique, programmation générative et métaprogrammation . . . 110
5.5.1 Choix techniques . . . 110
5.5.2 Programmation générique et factorisation de code . . . 111
5.5.3 Métaprogrammation . . . 114
5.6 Application à la cosmologie . . . 121
5.6.1 Vers des bibliothèques actives pour la cosmologie : l'exemple de MAGRATHEA . . . 121

5.6.2 Proposition d'architecture logicielle unifiée pour la cosmologie numérique . . . 127

6 Vers des orthtrees plus raffinés **131**
6.1 Position du problème . . . 132
6.1.1 Description d'une implémentation classique : le cas de RAMSES . . . 132
6.1.2 Limitations des implémentations traditionnelles . . . 133
6.2 Solution unifiée . . . 136
6.2.1 Principe de l'algorithme . . . 136
6.2.2 Propriétés locales et encodage des indices . . . 137
6.2.3 Propriétés globales et émergence de l'arbre . . . 145
6.2.4 Avantages et limitations . . . 148
6.3 Application et perspectives . . . 152
6.3.1 Implémentation et améliorations . . . 152
6.3.2 Vers une généralisation à tout type d'arbre . . . 153

7 Vers un code de raytracing en métrique faiblement perturbée **155**
7.1 Élements de conception . . . 156
7.1.1 Algorithme . . . 156
7.1.2 Architecture et implémentation . . . 160
7.1.3 Parallélisation . . . 161
7.2 Tests et vérifications . . . 164
7.2.1 Construction de l'arbre . . . 164
7.2.2 Intégration dans le cas homogène . . . 165
7.2.3 Intégration en métrique de Schwarzschild . . . 166
7.3 Application cosmologique . . . 172
7.3.1 Intégration des géodésiques dans les cônes de lumière du Full Universe Run . . . 172
7.3.2 Perspectives . . . 173

8 Conclusion **175**

A Développements autour de la gravité de Brans-Dicke **179**
A.1 Théorie de Brans-Dicke et hypothèse AWE . . . 179
A.2 Expression des actions . . . 179
A.3 Application à la cosmologie . . . 180
A.4 Solutions analytiques . . . 181
A.5 Solutions numériques . . . 183
A.6 Résultats . . . 185
A.7 Conclusion . . . 187

B Géodésiques nulles en métrique de Schwarzschild **189**
B.1 Expressions en termes d'intégrales . . . 189
B.2 Solutions en fonctions de Weierstrass . . . 193

C Articles et publications **195**
C.1 Solution elliptiques en gravité généralisée de Brans-Dicke avec un couplage non-universel 195
C.2 Oscillations acoustiques des baryons limitées par la variance cosmique au sein de la simulation DEUS-FUR ΛCDM . . . 196
C.3 Matrice de covariance du spectre de puissance de la matière dans les simulations DEUS-PUR ΛCDM : résolution en masse et erreurs non-gaussiennes . . . 196
C.4 La première simulation à l'échelle de l'Univers Observable . . . 197
C.5 DEUS Full Observable Universe Simulations : défi numérique et perspectives . . . 197

Avant-propos

***

« Depuis l'aube des temps, l'Homme n'a cessé de s'interroger sur l'origine de tout ce qui est et sur sa place au sein de l'Univers. Au fil des millénaires, en levant les yeux vers le cosmos, il a peu a peu percé les mystères du mouvement des astres, le conduisant aujourd'hui jusqu'à deux des plus grands problèmes ouverts de la physique moderne : la *matière noire* et l'*énergie noire*. Pour percer ces mystères, des instruments de plus en plus perfectionnés nous ont fait entrer dans une ère de *cosmologie de précision*, nous apportant une connaissance toujours plus fine des paramètres cosmologiques. Le *Big-Bang* qui s'est produit il y a environ 13.8 milliards d'années a ainsi marqué le point de départ de l'Univers tel que nous le connaissons aujourd'hui, composé à 69% d'énergie noire, à 26% de matière noire et à 5% de matière ordinaire. Dans les années à venir, et grâce à l'avènement du *Big-Data* et aux possibilités offertes par l'*Exascale*, l'analyse de relevés observationnels toujours plus grands et de simulations numériques toujours plus résolues nous permettra de repousser les limites du savoir et de déterminer, enfin, la nature de l'énergie et de la matière noire. » C'est à cette histoire convenue, lisse et sans accroc que ne vous *convie pas* le reste de cette thèse.

En effet, si c'est bien ces thèmes que la présente thèse se propose d'aborder, c'est avec une philosophie et une logique globale légèrement différentes qu'elle se propose de le faire. L'objectif des paragraphes qui suivent est donc de placer le propos au sein d'un contexte plus large, et de présenter succinctement la philosophie dans laquelle ce travail a été effectué en vue de lui garantir un maximum de clarté. Cela est d'autant plus important que ce travail se veut comme le plus interdisciplinaire possible entre la cosmologie et l'informatique. Ainsi, des questions de physique sur la structuration de l'Univers et sur la propagation de la lumière au sein d'un champ de matière inhomogène nous conduiront à traiter des questions de conception de langages et de métaprogrammation, non pour des aspects de nature purement technique, mais pour des raisons plus fondamentales en vue de proposer de nouvelles approches numériques quant à la résolution de problématiques théoriques.

La cosmologie scientifique moderne est une branche de la physique passionnante, non pour les raisons évoquées dans le premier paragraphe, mais parce que le parallèle avec l'œuvre de Thomas Kuhn, *La structure des révolutions scientifiques* est suffisamment troublant pour être relevé. Elle semble [1] en effet être entrée, ces dernières années, dans une phase prérévolutionnaire. La matière noire, l'énergie noire, le modèle de concordance ΛCDM et le fait même de parler d'ère de cosmologie de précision sont autant de signes pouvant être interprétés, sous l'œil de Kuhn, comme annonciateurs d'un profond bouleversement de cette science. La cosmologie scientifique moderne est donc passionnante parce qu'il se pourrait qu'elle assiste *hic et nunc* aux prémices d'un changement de paradigme [2]. À la fin du chapitre VII de *La structure des révolutions scientifiques*, l'auteur écrit : « Face à une anomalie ou à une crise, les scientifiques adoptent une attitude différente à l'égard des paradigmes existants et la nature de leurs recherches change en conséquence. La prolifération des variantes concurrentes du paradigme,

1. « Les prévisions sont difficiles, surtout lorsqu'elles concernent l'avenir. »
2. Dans toute la suite du texte, l'emploi du mot *paradigme* est à interpréter dans le cadre d'une acception kuhnienne et non dans le sens, légèrement pompeux, de « paradigme informatique ».

le fait d'être disposé à essayer n'importe quoi, l'expression d'un mécontentement manifeste, le recours à la philosophie et à des discussions sur les fondements théoriques, tous ces signes sont autant de symptômes d'un passage de la recherche normale à la recherche extraordinaire [3]. » C'est justement la présence de ces symptômes qui donnent aujourd'hui un caractère très particulier à la cosmologie et qui font d'elle un sujet d'intérêt pour des aspects non seulement scientifiques, mais également pour des aspects sociologiques de la recherche, pour des aspects d'histoire des sciences et pour des aspects épistémologiques.

Que tout cela ne soit pas interprété faussement. Il n'est ici ni question de tomber dans le piège d'un relativisme scientifique exacerbé, ni question de s'attaquer à un modèle particulier au seul titre qu'il est au cœur du paradigme actuel. Il s'agit juste de replacer ce paradigme dans un contexte plus large pour essayer de s'extraire, autant que faire se peut, de la subjectivité liée aux préférences de chacun vis à vis de tel ou tel modèle. Le fait que telle observation ou telle simulation renforce WCDM plutôt que ΛCDM ou RPCDM [4] grâce à la mesure d'un paramètre donné à la cinquième décimale ne doit pas être l'aboutissement de la recherche en cosmologie, mais le point de départ d'une réflexion sur la physique fondamentale favorisée par le modèle en question. S'il sera effectivement question dans la suite de matière noire et d'énergie noire, celles-ci ne devront pas être considérées comme autre chose que des hypothèses de travail : d'un point de vue théorique, elles ne sont qu'une approche ontologique parmi d'autres. Des approches de nature législative, comme peuvent l'être les théories de gravité modifiée, ou de nature paradigmatique, comme peuvent l'être les recherches sur la conjecture de la *backreaction*, sont au cœur de nombre de propositions alternatives. Et c'est justement ce foisonnement d'approches de natures profondément différentes qui trouve un écho dans l'œuvre de Kuhn. Quoi qu'il en soit, la matière noire et l'énergie noire n'ont jamais été, et ne seront jamais, comme on le lit parfois, les deux plus grands problèmes ouverts de la physique moderne, tout du moins au sens scientifique. Appeler une hypothèse un problème n'a que peu de sens. L'anomalie des courbes de rotation des galaxies spirales, l'anomalie de la luminosité des supernovæ, l'interprétation du spectre de puissance du fond diffus cosmologique, la structure du bullet cluster : là sont quelques-uns des problèmes ouverts de la cosmologie à l'aube du XXI^{e} siècle.

Bien entendu, les grands projets, qu'ils soient observationnels, expérimentaux ou numériques ont un rôle de première importance à jouer dans la résolution de ces problèmes ouverts. Mais il faudrait bien se garder de dire qu'une expérience a parlé de manière définitive : une expérience ne « parle » jamais, on lui fait dire des choses, ce qui est complètement différent. Ou, dit autrement, les résultats d'une expérience sont toujours munis d'une gangue d'interprétation. L'expérience est ce par quoi nous sommes capables de faire « cracher » des nombres à la Nature au sein d'un cadre théorique donné : oui l'Univers est constitué à 69% d'énergie noire et à 31% de matière ... si tant est qu'il soit réellement homogène et isotrope à grande échelle et que Ω_Λ et Ω_m soient bien des paramètres fondamentaux de la cosmologie. En ce sens, les résultats observationnels, expérimentaux ou numériques sont capitaux, non parce qu'ils nous renseignent sur la valeur exacte des paramètres, mais parce qu'ils sont susceptibles de mettre en tension leur propre cadre d'interprétation, nous permettant ainsi de mieux comprendre ce que nous mesurons réellement.

Quelle place doit donc avoir le numérique dans un tel contexte ? Traditionnellement, l'astrophysique naissait du dialogue sans cesse renouvelé entre théorie et observation, cette dernière bénéficiant d'outils toujours plus évolués pour questionner la Nature par le biais de l'instrumentation. Théorie et observation n'étaient donc pas sur un pied d'égalité dans le sens où l'on prenait presque systématiquement l'observation, enveloppée de son tissu d'instrumentation, comme point de départ factuel pour interroger et remettre en cause la théorie d'une manière toujours plus précise. La simulation numérique est venue rétablir le nécessaire équilibre de ce dialogue en fournissant à la théorie les outils techno-

3. À comprendre dans le sens de recherche non-incrémentale.
4. Ces modèles cosmologiques seront explicités dans la suite du manuscrit.

logiques dont elle avait besoin pour venir réinterroger les observations d'une manière toujours plus pertinente. Ainsi, si l'expérience permet, comme dit précédemment, de « faire cracher des nombres » à la Nature, la simulation numérique permet de « faire cracher des nombres » aux lois physiques. Ou de manière plus imagée, si l'observation et son interprétation est ce qui permet d'extraire les intelligibilités du monde empirique complexe et changeant, la simulation numérique est ce qui permet de déconstruire les symétries des théories physiques. Il s'agit là d'un point capital : le physicien et le numéricien ont, de fait, des approches antagonistes. En effet, si le rôle du premier est de construire des symétries au niveau le plus fondamental, le rôle du second est de les déconstruire de la façon la plus méthodique possible. Cette thèse essaye de s'inscrire au confluent de ces deux modes de pensée et s'interroge, entre autres choses, sur les approches numériques qui seront le plus à même d'explorer le foisonnement de propositions théoriques résultant du « fait d'être disposé à essayer n'importe quoi » si la cosmologie est bel est bien entrée dans une phase prérévolutionnaire au sens kuhnien du terme. Le travail interdisciplinaire présenté dans ce manuscrit s'interroge donc sur les méthodes qui nous permettrons d'être en mesure, dans les années à venir, d'utiliser la puissance offerte par l'outil numérique pour tester la pertinence de multiples propositions ontologiques, législatives et paradigmatiques.

Mais si la simulation numérique a ouvert un nouvel espace des possibles, elle n'en reste pas moins une méthode très jeune du point de vue de la démarche scientifique. Il est donc également important de s'interroger sur la place qu'elle doit trouver au sein d'un processus de connaissance et ainsi fixer les frontières de ce qu'elle peut et de ce qu'elle ne peut pas réaliser. Sans quoi, elle risquerait d'outrepasser ses limites, promettrait l'impossible et serait à même de mettre à mal la nécessaire rigueur de toute entreprise scientifique. La puissance prédictive de la physique provient entre autres choses du renoncement face à la complexité de la Nature : plutôt que de chercher à expliquer le réel dans son entièreté, le physicien, dans le cadre de la démarche scientifique, s'emploie à isoler quelques cas simples pour remonter plus facilement aux principes et aux lois fondamentales qui semblent sous-tendre la réalité. De la même manière, pour maximiser sa pertinence du point de vue scientifique, la simulation numérique doit renoncer à sa promesse de « tout calculer » et se limiter à quelques cas d'utilisation bien définis. À cette condition, et à cette condition seulement, elle peut devenir un outil d'une redoutable efficacité. C'est entre autres pour cette raison, ajoutée à l'absence de cosmologie de laboratoire à même de reproduire les phénomènes physiques modélisés dans une large gamme de conditions, que les modélisations semi-analytiques et les modèles sous-grilles seront évités autant que faire se peut de manière à garantir la plus grande lisibilité possible aux résultats.

À cette réflexion à mi-chemin entre la cosmologie et l'informatique, viennent se greffer deux domaines en pleine ébullition, le « Big-Data » et la course à l'« Exascale ». Quand bien même il s'agirait de modes passagères, ces deux thématiques se trouvent être au confluent d'une lente et profonde mutation de la méthode scientifique. La première d'entre elle, le Big-Data, est liée à l'accumulation grandissante de données scientifiques de tous types, notamment par les grands relevés observationnels et par les *Grand Challenge*[5] numériques. C'est ainsi que des missions spatiales comme GAIA et EUCLID nous livreront des cartographies précises et méthodiques de la distribution d'étoiles dans la galaxie, et de la distribution de galaxies dans l'Univers Observable et que d'autres projets internationaux comme le Square Kilometer Array annoncent dès à présent qu'ils produiront plus d'un exaoctet de données astrophysiques brutes par jour. Cela n'est pas sans poser d'immenses problématiques techniques et de grandes questions quant à la valorisation et au partage des données scientifiques. Car si parvenir à réduire et post-traiter ces données est une chose, être capable d'en exploiter toutes les potentialités et ainsi éviter de fabriquer de « Big » cimetières de données en est une toute autre. Toutefois la véritable question n'est pas là. En effet, c'est la problématique même de notre rapport au savoir que vient réinterroger le Big-Data. Non parce qu'il est le prolongement naturel de 400 ans de démarche scientifique, mais parce qu'il s'inscrit, au contraire, dans une logique complètement différente. Le Big-Data,

5. Terme consacré aux expériences numériques de grande ampleur durant lesquelles une partie significative d'un supercalculateur peut être réservé à une unique simulation.

c'est la phénoménologie rendue prédictive, et c'est à ce titre qu'il serait possible d'y voir la naissance d'une nouvelle façon de connaître. Depuis Galilée, la physique, dans son acception moderne, a procédé essentiellement d'une démarche inductive visant à isoler les lois fondamentales de la Nature et d'une démarche déductive visant à formuler des prédictions falsifiables à partir de ces lois fondamentales. Dans ce cadre, les lois fondamentales de la Nature acquièrent, de fait, le statut d'explication : expliquer c'est déterminer l'axiomatique physique la plus simple et la plus générique possible capable de décrire le monde, c'est-à-dire capable de formaliser les relations de causalité entre évènements, et à partir de laquelle tout le corpus de la physique peut être démontré et vérifié par l'expérience. Mais les lois physiques dont il s'agit ici vont bien plus loin qu'une simple mise en équation de la Nature dotée d'une redoutable efficacité prédictive : elles nous interrogent sur ce qu'est la réalité, non moins pour acter de ce que le monde n'est pas, que pour faciliter l'émergence de nouveaux paradigmes. Dit autrement, les lois fondamentales sont ce par quoi nous nous forgeons une représentation du monde. Et le Big-Data peut tout à fait trouver sa place dans une telle logique inductivo-déductive, la mise en évidence de corrélations pouvant servir de base à la recherche de relations de causalité entre évènements, apportant ainsi de nouveaux éléments quant à l'identification de lois sous-jacentes.

Mais parallèlement à cette approche classique, inscrite dans le prolongement de l'histoire de la physique galiléenne, le Big-Data rend possible une autre démarche, visant à prévoir sans expliquer grâce à la seule force des statistiques et de l'accumulation des données. Historiquement, cette approche a prévalu pendant des centaines d'années pour l'analyse des systèmes complexes, et le développement de la médecine en fut la parfaite illustration. Certes la chimie, la biologie moléculaire et la génétique ont permis de faire des progrès considérables quant aux traitements médicaux, mais ce n'est pas parce que nous ne disposions pas des connaissances relatives à ces thématiques que nous étions dans l'incapacité totale de soigner. Toutefois, dans un cadre scientifique traditionnel, la mise en évidence et la compréhension de processus plus fondamentaux allait presque systématiquement de pair avec une amélioration de la capacité prédictive de la thématique considérée. Et c'est justement ce point que vient réinterroger le Big-Data d'une manière radicalement nouvelle. Car l'accumulation de données expérimentales hétérogènes, couplées d'une part à une capacité de traitement de l'information unique dans l'histoire et d'autre part à un cadre statistique mathématiquement solide rend possible la prédiction sans avoir recours à l'explication par des lois plus fondamentales. C'est ainsi qu'après avoir passé des années à essayer d'enseigner à des programmes informatiques la construction d'un langage et de sa grammaire, il s'est avéré bien plus efficace d'utiliser des approches statistiques pour traduire des textes d'une langue à une autre[6]. Mais alors, s'il n'est plus nécessaire d'expliquer les phénomènes par des processus de plus en plus fondamentaux pour prévoir ces mêmes phénomènes, ne serait-ce pas le sens même d'explication qu'il conviendrait de redéfinir ? C'est justement pour cette raison que nous assistons, peut-être, à l'émergence d'une nouvelle démarche de connaissance, parallèle et complémentaire à la démarche scientifique traditionnelle. Ou, au lieu de nouvelle, il serait sans doute plus juste de dire que l'informatique et la statistique sont en train de donner à une approche phénoménologique séculaire une capacité prédictive que seule la démarche scientifique héritée de Galilée semblait à même d'atteindre. Il ne s'agit pas de rejeter une telle démarche au seul titre qu'elle ne permettrait pas d'accéder à une compréhension aussi fine de la nature des phénomènes qu'une démarche inductivo-déductive : après tout, certes la démarche scientifique basée sur une mise en équation de la Nature fonctionne, mais la raison profonde de son efficacité demeure, à ce jour, complètement inconnue. La différence majeure vient peut être du fait que cette dernière, en mettant au jour des lois de plus en plus fondamentales, fournit un cadre d'interprétation de la réalité et s'accompagne donc périodiquement de changement de paradigmes, au gré des révolutions scientifiques. Ce que ne peut fournir une démarche de connaissance capable de prédire le comportement de systèmes complexes sans en avoir pour autant déchiffré les mécaniques internes. Il est à noter également que dans de telles approches, la donnée est en général élevée au rang de mesure objective. Or, en physique, ce n'est presque jamais le cas : une mesure ne sau-

6. C'est tout du moins l'approche aujourd'hui adoptée par la grande majorité des services de traduction en ligne pour des questions de pertinence et d'efficacité.

rait être objective puisqu'elle est presque toujours réalisée au sein d'un cadre d'interprétation donné. Aussi, mesurer le redshift pour tenter de mieux connaître la cosmologie, c'est déjà attribuer un sens au décalage spectral dans un cadre théorique donné. Pour toutes ces raisons, et parce que la démarche de connaissance que semble rendre possible le Big-Data est encore jeune au regard de la démarche scientifique traditionnelle, cette thèse se limitera à cette dernière et l'accumulation de données ne sera utilisée pour produire des modèles semi-analytiques visant à prédire sans nécessairement expliquer. Et c'est même la démarche inverse qui sera le plus souvent adoptée, ou tout du moins visée : celle qui a trait à la notion d'émergence.

Cette notion d'émergence, si elle n'est pas conceptuellement nouvelle, connait une profonde mutation du fait de l'accroissement de la capacité de calcul. Mais pour le comprendre, il est nécessaire de se replacer dans le contexte de l'utilisation du calcul haute performance en physique et plus spécifiquement en cosmologie. À chaque fois qu'une nouvelle machine est déployée, deux approches, correspondant à deux philosophies différentes, sont généralement mises en œuvre pour utiliser ces moyens de calcul. La première vise à accroître la résolution, la statistique et l'exploration de l'espace des paramètres à physique donnée : de façon schématique, il s'agit d'en comprendre davantage à partir des mêmes ingrédients physiques. Le projet Dark Energy Universe Simulation : Full Universe Run dont il sera question dans toute une partie de ce travail s'inscrit en ligne directe de cette démarche. La seconde approche consiste à explorer une plus large gamme de phénomènes physiques. Inclure la physico-chimie des nuages interstellaires, l'explosion de supernovæ, les phénomènes à haute énergie au sein de simulations hydrodynamiques s'inscrit dans cette optique. Toutefois, à trop multiplier les ingrédients, les paramètres libres et les modèles sous-grille, c'est le caractère prédictif de la physique que l'on vient mettre en tension, et ce, quelle que soit l'étendue des phénomènes que l'on semble à même de reproduire au sein des simulations. Car, encore une fois, il est tout à fait possible de reproduire des systèmes complexes sans pour autant les comprendre. Il ne s'agit pas ici d'un jugement quant à la validité de telle ou telle approche, il s'agit juste de mettre en lumière deux philosophies différentes liées à l'utilisation des ressources de calcul. Mais il existe une troisième possibilité, qui devrait être, du moins du point de vue de l'auteur de cette thèse, l'essence même du calcul haute performance en physique dans une logique cartésienne : le calcul *ab initio*, c'est-à-dire le calcul « à partir des principes fondamentaux ». En d'autres termes, il s'agit de profiter d'un accroissement de la capacité de traitement de l'information non pas pour étudier plus de phénomènes physiques dans un déferlement de modèles empiriques, mais pour étudier les mêmes phénomènes physiques sous un angle différent, en utilisant une physique de plus en plus épurée, de plus en plus minimaliste et en partant de lois de plus en plus fondamentales. Il s'agit, autrement dit, de mettre en évidence le caractère émergent d'un certain nombre de théories physiques. Mettre en lumière et mieux comprendre l'émergence de la structuration de l'Univers grâce à des simulations cosmologiques est un premier pas intéressant, pertinent et nécessaire, mais ce n'est pas de cela dont il serait question, notamment dans un contexte où l'on disposerait de machines exascale. Cela pourrait et devrait aller beaucoup plus loin.

Car la cosmologie n'est et ne saurait être une théorie physique fondamentale. Le principe cosmologique, hérité d'une vision copernicienne tout à fait à propos, s'il semble conceptuellement raisonnable et apporte à la cosmologie analytique les symétries dont elle a besoin pour s'exprimer, n'en reste pas moins une hypothèse *ad hoc*. Non parce qu'il n'est pas essentiel, mais parce qu'il serait démontrable et vérifiable par une approche *ab initio*. L'Exascale ne promet rien de moins que de nous offrir la possibilité de déconstruire méthodiquement la cosmologie enchâssée dans son socle de symétries pour la reconstruire à partir de la physique sous-jacente, c'est-à-dire la relativité générale. En d'autres termes, c'est de bâtir une cosmologie sans modèles mais à partir des seules lois fondamentales exprimées dans leur plus large généralité dont il s'agit ici. Dans un tel cadre, la question des conditions initiales est évidemment non triviale bien qu'en principe, rien ne s'oppose à la démontrabilité du caractère émergent de la cosmologie physique à partir de la relativité générale seule. La portée d'une telle démonstration, si elle était réalisée, irait bien plus loin que la seule confirmation de la validité d'un siècle de cosmologie

analytique. Elle viendrait, entre autres choses, confirmer ou infirmer le principe cosmologique, tester les effets de la relativité générale à grande échelle, et questionner d'une nouvelle manière les problèmes des courbes de rotation des galaxies et de l'accélération de l'expansion de l'Univers. Mais une cosmologie sans principe cosmologique est-elle au moins possible ? Poser cette question, c'est poser la question de la nécessité de lois physiques globales de l'Univers. Et rien, dans notre compréhension actuelle de la physique, ne semble motiver une telle nécessité. Une cosmologie sans principe cosmologique, c'est une cosmologie qui prend acte du fait qu'elle n'est que la moyenne globale d'effets physiques fondamentalement locaux, de la même manière que l'évolution du climat d'une planète n'est rien d'autre que la résultante globale de mécanismes climatiques locaux. Ce faisant, l'interprétation des observations devient certes beaucoup plus subtile, mais rien n'impose qu'il devrait en être autrement, sinon l'ajout traditionnel de symétries *ad hoc*. Quoi qu'il en soit, la question n'est pas ici de se prononcer en faveur de telle hypothèse ou de telle autre, la question est de s'interroger sur l'usage le plus pertinent d'une capacité de calcul accrue dans un contexte de course à l'Exascale. Et venir réinterroger les bases fondamentales de la cosmologie physique au moyen de nouveaux outils dans une période potentiellement prérévolutionnaire semble particulièrement à propos.

Alors comment faire ? Comment mettre en évidence, dans la pratique, le caractère émergent de la cosmologie physique ? Il ne s'agit pas d'une question accessoire, mais au contraire d'une authentique problématique fondamentale à long terme comportant un certain nombre de verrous tant théoriques que numériques. Car actuellement, et peut être étonnament aussi, du moins d'un point de vue extérieur à la communauté cosmologique, le terrain d'une cosmologie numérique réellement relativiste, c'est-à-dire sans autre ingrédient que des conditions initiales adéquates et les équations de la relativité générale, semble aussi lointain qu'inexploré. Dans ce cadre, parvenir à traiter le problème newtonien de formation des grandes structures dans son ensemble, prendre conscience des problématiques numériques soulevées par celui-ci pour ouvrir la voie à des codes exascale plus simples, plus génériques et plus performants, et aborder les problèmes cinématiques puis dynamiques en métrique perturbée peuvent être vue comme autant d'étapes qui prépareront finalement le saut conceptuel que représenterait un code dynamique relativiste pour la cosmologie. Et si cette thèse ne saurait prétendre à attaquer ce dernier problème frontalement, c'est bien d'amener des éléments de réflexion concernant les premiers points dont il s'agit ici.

Ainsi, deux problématiques interdépendantes seront abordées : d'une part la question de la propagation de la lumière au sein d'un univers structuré en expansion accélérée et d'autre part la question de la mise en place de nouvelles approches numériques pour la cosmologie. Car l'un des meilleurs moyens de s'interroger sur ce que nous disent vraiment les observations sur l'évolution et la dynamique de l'Univers, c'est peut-être encore d'étudier ce que verrait un observateur virtuel au sein d'un univers numérique. C'est tout du moins l'approche qui sera mise en œuvre ici, y compris si cela doit passer par la réalisation de simulations de la taille de l'Univers Observable pour minimiser les approximations et les hypothèses utilisées au cours des analyses. De la même manière, et dans un esprit *ab initio*, la propagation de la lumière des grandes structures jusqu'à l'observateur sera réduit à sa substantifique moelle, évitant ainsi un maximum d'approximations. Le redshift, le lentillage gravitationnel, l'effet Sachs-Wolfe intégré, le calcul de la distance de diamètre angulaire et de la distance luminosité sont autant de facettes d'un seul et unique phénomène fondamental : la propagation de photons suivant les équations de la relativité générale. C'est donc au problème de la détermination des géodésiques nulles en cosmologie que sera consacrée toute une partie de ce manuscrit, non moins parce que cela permet de traiter le problème de l'Univers observé d'une façon nouvelle que parce que cela ouvre la voie vers un nombre significatif d'analyses ayant trait à des observables souvent considérées comme indépendantes. Il s'agira donc de s'appuyer sur des simulations cosmologiques de dernière génération pour y introduire un traitement relativiste consistant, c'est-à-dire un traitement relativiste de la cinématique en métrique faiblement perturbée. Et quand on sait combien la relativité générale peut impacter les relevés observationnels, ne serait-ce que par les phénomènes de lentillage, la question des contreparties

dynamiques de ces effets n'en devient que plus intéressante. C'est entre autres choses dans le but d'être un jour en mesure de traiter de tels effets, que la production de cônes de lumière et que le traitement de la question de l'intégration numérique des équations géodésiques au sein des simulations cosmologiques nous conduiront à mener une réflexion de fond quant à la conception des codes numériques. La question n'est donc pas de traiter le calcul des géodésiques nulles comme un problème isolé, mais plutôt comme faisant partie d'un projet à plus long terme visant d'une part à explorer un foisonnement théorique nécessitant un nombre toujours plus grand de contraintes observables et d'autre part à poser les bases d'une architecture logicielle qui permettrait le développement d'une cosmologie *ab initio* dans le contexte de l'Exascale.

Aussi, le présent manuscrit résume les résultats de trois ans d'un travail de thèse s'inscrivant dans cette logique. Ce travail, marqué dès le départ par une forte interdisciplinarité scientifico-numérique, a débuté par un questionnement autour des cônes de lumières au sein des simulations cosmologiques. Il s'agissait de profiter de l'opportunité offerte par le Full Universe Run [7] pour implémenter la construction d'univers observables virtuels, incluant d'une part les informations relatives à la formation des grandes structures et d'autre part, et pour la première fois, les informations relatives au potentiel gravitationnel. Cela allait rendre possible la création de plusieurs espaces des redshifts pour différents observateurs et pour différents modèles cosmologiques en évitant un certain nombre d'approximations usuelles [8]. La réalisation des Runs en tant que Grand Challenge sur la machine Curie entre février et août 2012 a constitué un travail d'équipe significatif pour mettre en place la chaîne de codes, les optimiser, les configurer de façon optimale pour utiliser l'ensemble de la machine, assurer le bon déroulement des simulations sur un grand instrument de calcul, gérer les données, les post-traiter et les archiver. Il s'agissait, en ce sens, d'un véritable défi numérique, poussant dans leurs derniers retranchements les codes de simulation sur l'une des machines les plus puissantes du monde au moment de son ouverture. Toutefois, la prouesse technique ne doit en aucun cas occulter la démarche scientifique dans le cadre de laquelle elle s'inscrit. Car, une fois encore, réaliser des simulations pour un ensemble de paramètres cosmologiques donné n'est pas une fin en soi, du moins ici. Il reste encore beaucoup trop de zones d'ombres pour qu'une simulation soit à même de valider définitivement un modèle cosmologique à l'heure actuelle. ΛCDM, WCDM ou RPCDM, là n'est donc pas la question : ce ne sont de toute façon ni des théories, ni des lois fondamentales, ce ne sont que des modèles et ils n'ont donc, par construction, pas vocation à fournir une ultime description de la réalité physique. La question n'est donc pas principalement de contraindre la cosmologie mais plutôt de comprendre comment la contraindre. Elle est de comprendre ce que nous mesurons réellement lorsque nous effectuons des observations. Elle est enfin, et peut être plus fondamentalement, de comprendre ce que la formation des structures a à nous dire sur la cosmologie et *vice versa*. C'est justement parce que la comparaison de l'évolution des grandes structures pour différentes cosmologies et pour différents observateurs sur le volume de l'Univers Observable est susceptible d'entraîner de nouveaux questionnements que le Full Universe Run est scientifiquement pertinent. Et c'est pour tenter d'apporter de nouveaux éléments de réflexion sur ces questions que s'en est suivi un travail analytique sur les géodésiques en formalisme faiblement perturbé. Le but était de fournir un cadre théorique compatible avec les approximations utilisées dans les codes tout en restant aussi proche que possible de la relativité générale pour analyser les simulations et mieux comprendre les multiples effets liés à la propagation de la lumière au sein des grandes structures. Parallèlement à cette étude, les leçons numériques tirées du Full Universe Run, puis du Parallel Universe Run destiné, entre autres choses, à tester la validité de l'hypothèse ergodique [9] allaient fournir les bases d'une réflexion de fond quant à la conception des codes en cosmologie. Car si les expériences numériques précédemment citées ont montré les succès des codes d'aujourd'hui elles ont aussi mis en exergue leurs limites. Limites techniques de scalabilité, d'optimisation, de gestion du

7. Il s'agit du premier ensemble de simulations cosmologiques à l'échelle de l'Univers Observable.
8. Il est par exemple fréquent de construire des cônes de lumière en répliquant les données lorsque les simulations ne couvrent pas un volume assez grand[1, 2].
9. C'est-à-dire tester l'équivalence des résultats statistiques donnés par une seule grande simulation et ceux calculés à partir d'un grand nombre de simulations de taille plus modeste.

cache et d'empreinte mémoire mais aussi et surtout limites de complexité et d'exploration de l'espace des possibles. S'aventurer avec ces codes dans les contrées de la physique au-delà du modèle de concordance ou se lancer avec eux sur la piste d'une cosmologie *ab initio* est une tâche difficile, et sans doute trop difficile. En effet, l'intrication des problématiques de parallélisation, de méthodes numériques et de physique dressent un mur de complexité qu'il devient de plus en plus difficile de franchir. Et si cette apparente complexité n'était qu'artificielle ? Car, au fond, les briques de base des codes de cosmologie sont toujours les mêmes : des intégrateurs, des arbres, des maillages, des particules, des quantités eulériennes, et quelques équations physiques. Là encore, et de façon presque ironique, la complexité n'est qu'une propriété émergente de l'assemblage de composantes fondamentales : la multiplicité des arrangements des briques de base entraîne mécaniquement une explosion combinatoire du nombre de lignes de code, et donc de la complexité de ces derniers. Et c'est au physicien numéricien que revient naturellement la gestion de cette complexité émergente. À moins que... À moins que tout ceci ne soit qu'un faux problème et que le rôle du physicien numéricien soit au contraire de se concentrer sur les briques de base, là où résident les véritables problématiques physiques et algorithmiques. Car si le rôle du physicien est d'identifier les symétries, les régularités, et les lois fondamentales pourquoi le numéricien devrait-il se refuser à adopter une approche similaire sur ses propres outils pour déconstruire ces mêmes symétries ? Assembler les pièces du puzzle et faire émerger la complexité ne devrait-il pas être simplement ... le travail des compilateurs ? Et dès lors, au lieu de chercher à « tordre » la physique pour l'exprimer au sein d'un langage donné, ne serait-il pas plus à propos de déformer les langages informatiques eux-mêmes pour que la physique puisse s'y exprimer librement ?

Toutes ces questions ont été au cœur du développement expérimental de la bibliothèque MAGRATHEA destinée en premier lieu à traiter le problème de la propagation de la lumière au sein du Full Universe Run et qui a constitué une part conséquente du travail de cette thèse. Mais ce n'est pas seulement de cela dont il s'agissait : c'est vraiment de tirer les leçons des simulations Grand Challenge pour repenser la conception des codes dans une optique de cosmologie *ab initio* sur de futures architectures exascale dont il était question. Et c'est dans ce cadre aussi, et parce que les maillages sont parmi les éléments les plus structurants des simulations cosmologiques, que s'est inscrit le développement d'orthtrees [10] plus génériques et performants. Tous ces aspects, tant analytiques que numériques ont conduit à la conception d'un code de raytracing massivement parallèle capable d'analyser la propagation de la lumière pour un champ gravitationnel arbitraire compatible avec une métrique faiblement perturbée, et donc capable de traiter les géodésiques nulles des observateurs virtuels du Full Universe Run. Cette approche diffère notablement des traitements usuels du weak-lensing en ce qu'elle ne repose que sur une distribution de potentiel et un traitement relativiste des trajectoires des photons sans approximation supplémentaire. Elle ouvre donc la porte à une exploration unifiée des problématiques de lentillage, d'effet Sachs-Wolfe intégré, de calcul des distances cosmologiques et plus largement tout ce qui repose sur la manière dont se propage la lumière aux grandes échelles. C'est sur la question des distances que se focalise le dernier chapitre de la première partie de cette thèse car elle est au cœur même de la détermination des paramètres cosmologiques. Il ne s'agira pas de mener une étude exhaustive de la question, mais de jeter quelques pistes susceptibles, à terme, de conduire à une meilleure compréhension des limites des relations existant entre la mesure des distances et la cosmologie. Et dans un contexte où les relevés observationnels promettent de sonder toujours plus finement la distribution de matière à grande échelle, comprendre exactement comment les photons se frayent un chemin des galaxies aux miroirs des télescopes au sein des grandes structures semble primordial. Non principalement parce que la détermination de ces effets permettrait de corriger les mesures, mais parce que l'acte même de mesurer l'Univers au moyen de la lumière porte en lui une information quant à la cosmologie sous-jacente. À l'heure d'une cosmologie de précision, et parce qu'une observation ne saurait être objective tant elle est conditionnée par le cadre théorique qui lui donne sa pertinence, il apparait donc comme de première importance de s'interroger sur ce que nous mesurons réellement. Et c'est tout du

10. Aussi appelés hyperoctrees, il s'agit de la généralisation en dimension arbitraire d'arbres spatiaux binaires couramment utilisés pour les maillages adaptatifs.

moins dans cet esprit qu'a été effectué ce travail de thèse.

Synthétiser un travail interdisciplinaire exhibant des aspects pouvant être considérés comme très techniques selon qu'on les considère sous l'œil du physicien ou sous l'œil de l'informaticien n'est pas chose aisée. Aussi, pour assurer une clarté maximale au propos, un plan en deux parties distinctes s'interpellant mutuellement a été choisi. Il s'agit d'un découpage à postériori, car le cheminement, lui, ne faisait pas cette distinction tant les problématiques étaient parfois intriquées, à l'image de ce qui a été décrit dans les paragraphes précédents. La première grande partie traitera exclusivement des aspects de cosmologie physique. Dans le premier chapitre, il s'agira de présenter succinctement le contexte de la cosmologie homogène : la plupart de ces résultats étant bien connus et la littérature étant abondante, le sujet n'est pas ici de reprendre cent ans d'histoire des sciences, mais plutôt de poser en quelques pages les bases des modèles et les conventions utilisées dans toute la suite. On y parlera donc de relativité générale, de matière noire, d'énergie noire, de modèle de concordance et l'on évoquera certaines extensions sur lesquelles un travail en marge du sujet principal de cette thèse a été effectué [11]. Le second chapitre portera sur la structuration de la matière au sein de l'Univers. Quelques éléments de base du scénario hiérarchique de formation des structures y seront résumés et des résultats physiques issus du projet Dark Energy Universe Simulation y seront présentés, notamment au sujet des oscillations acoustiques des baryons qui constituent un élément fondamental de la détermination des échelles de distance dans l'Univers. C'est dans ce chapitre que sera discutée la pertinence de quelques observables cosmologiques et que la caractérisation d'une cosmologie par les structures qu'elle engendre sera abordée. La question des observables nous conduira naturellement vers la problématique de la propagation de la lumière qui sera traitée d'un point de vue analytique au chapitre trois. Il s'agira d'étendre le premier chapitre à l'aide de la théorie des perturbations cosmologiques pour exhiber la forme des géodésiques pour une métrique faiblement perturbée compatible avec les observations et les simulations. À la fin de ce chapitre, l'ensemble des équations à résoudre numériquement pour traiter le problème de la propagation de la lumière à partir du seul potentiel gravitationnel sera donc établi. Le quatrième chapitre conclura cette première partie en se concentrant sur le problème de la mesure des distances cosmologiques et en présentant les résultats de l'application du formalisme exposé au chapitre précédent.

La seconde partie, quant à elle, traitera des aspects de cosmologie numérique. Le cinquième chapitre comprendra ainsi une introduction rapide à ce domaine, notamment au travers de l'exemple des diverses composantes du projet Dark Energy Universe Simulation. Le Full Universe Run permettra de présenter les succès mais aussi les limites des codes de simulation actuels, tant en termes de performances qu'en termes de conception. Le cœur de ce chapitre cinq tentera d'apporter des solutions à ces questions : on y abordera les concepts de programmation générique, de métaprogrammation, de bibliothèques actives et de langages embarqués avec comme objectif de proposer de nouvelles approches pour traiter toujours plus de pistes théoriques et pour poser les bases d'une cosmologie *ab initio*. Dans le chapitre suivant, un élément particulier, présent dans la très grande majorité des codes cosmologiques, sera étudié : les orthtrees. Un nouvel algorithme de raffinement adaptatif à la fois plus simple, plus générique et plus performant que ceux traditionnellement utilisés en cosmologie sera introduit. Enfin, en s'appuyant sur une nouvelle vision de la conception de codes et sur ce nouvel algorithme de raffinement adaptatif, le dernier chapitre traitera de la question de la propagation de la lumière en approximation de champ faible d'un point de vue numérique. Autrement dit, il portera sur l'élaboration et la validation du code de raytracing permettant d'aboutir aux résultats physiques présentés au quatrième chapitre. La conclusion du manuscrit dressera un bilan de l'ensemble et présentera des ouvertures aux diverses pistes d'études abordées dans cette thèse.

C'est ici que s'arrête cette longue introduction, place maintenant à la cosmologie physique et numérique, et bonne lecture !

11. Il est ici question d'un travail sur certains modèles liés aux théories de Brans-Dicke de la gravitation.

Chapitre 1

L'Univers homogène : modèle de concordance et extensions

***

1.1 Élements de relativité générale et métrique FLRW 2
1.1.1 Équation de champ d'Einstein 2
1.1.2 Équations de Friedmann-Lemaître 3
1.1.3 Contenu de l'Univers 4
1.1.4 Quantités cosmologiques 6
1.2 Modèle de concordance ΛCDM 8
1.2.1 Élaboration du modèle 8
1.2.2 Matière noire 9
1.2.3 Énergie noire 11
1.2.4 Propriétés dynamiques 13
1.3 Au-delà de la constante cosmologique 16
1.3.1 Validité du modèle de concordance et alternatives 16
1.3.2 Quintessence et autres équations d'état 18
1.3.3 Développements autour de la gravité de Brans-Dicke 19

Étudier un système physique nécessite d'abord de fixer le cadre théorique au sein duquel on se place, les conventions et les hypothèses utilisées, et la cosmologie ne saurait déroger à cette règle. Aussi, l'objectif de ce chapitre introductif est de dessiner succinctement les contours de la cosmologie standard, de ses hypothèses, et de ses extensions qui constitueront la base de travail du reste de cette thèse. Il n'est en aucun cas question de se lancer dans un cours exhaustif sur le modèle de concordance de la cosmologie homogène : cela n'est pas l'objet, et la documentation sur le sujet est déjà abondante [3, 4]. Il s'agit plutôt, dans un souci de clarté, d'en résumer les fondements théoriques et les hypothèses. Il s'agit aussi de présenter le *background* homogène qui sera perturbé dans les chapitres ultérieurs. Dans cette optique, la première partie de ce chapitre se concentrera sur la relativité générale et la métrique de Friedmann-Lemaître-Robertson-Walker. La seconde abordera le modèle ΛCDM et notamment deux de ses composantes principales : la matière noire et l'énergie noire. Enfin, dans la troisième, il sera question de certaines extensions et alternatives : WCDM, RPCDM, mais aussi de certains développements autour de la théorie de Brans-Dicke.

1.1 Élements de relativité générale et métrique FLRW

1.1.1 Équation de champ d'Einstein

Principe

S'il est une théorie physique que l'on pourrait qualifier d'élégante, c'est bien la relativité générale. Pour les mathématiques qu'elle met en œuvre bien entendu, mais aussi et surtout pour la révolution philosophique qu'elle a entraîné à partir de principes relativement simples. L'un de ces piliers est le principe d'équivalence, qui existe sous trois formes :

- le principe d'équivalence faible
- le principe d'équivalence d'Einstein
- le principe d'équivalence fort

La relativité générale d'Einstein, formulée en 1915, et la théorie de la gravitation de Nordström, aujourd'hui invalidée, semblent être les seules théories métriques de la gravitation à même de respecter le principe fort : pour ces théories, la gravitation est entièrement décrite par la géométrie de l'espace-temps $g_{\mu\nu}$. Ce n'est pas le cas des théories scalaire-tenseurs de la gravitation telle que la théorie de Brans-Dicke pour laquelle seule le principe d'équivalence d'Einstein est vérifié : si ce dernier a pour l'instant été validé à des précisions de l'ordre de 10^{-13}[5], les tests expérimentaux du principe fort restent donc de première importance pour apposer des contraintes sur toute une classe de théories de la gravitation.

Formulation

Mathématiquement, la relativité générale peut s'exprimer sous la forme de l'équation de champ d'Einstein qui établit une égalité locale entre le tenseur d'Einstein $G_{\mu\nu}$ décrivant la géométrie de l'espace-temps et le tenseur énergie-impulsion $T_{\mu\nu}$ décrivant son contenu énergétique :

$$\boxed{G_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu} \qquad \text{avec} \qquad G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2} g_{\mu\nu} R} \tag{1.1}$$

où :

- $g_{\mu\nu}$ est le tenseur métrique, avec $g^{\mu\sigma} g_{\sigma\nu} = g_{\nu\sigma} g^{\sigma\mu} = \delta^{\mu}_{\nu}$ et $\mathrm{d}s^2 = g_{\mu\nu}\,\mathrm{d}x^{\mu}\,\mathrm{d}x^{\nu}$
- $R_{\mu\nu} = g^{\sigma\rho} R_{\mu\sigma\nu\rho}$ est le tenseur de Ricci, correspondant à la trace du tenseur de Riemann
- $R = g^{\mu\nu} R_{\mu\nu}$ est la courbure scalaire, correspondant à la trace du tenseur de Ricci
- $G \approx 6.673\,84 \times 10^{-11}\ \mathrm{m^3 \cdot kg^{-1} \cdot s^{-2}}$ est la constante universelle de gravitation
- $c = 299\,792\,458\ \mathrm{m \cdot s^{-1}}$ est la vitesse de la lumière dans le vide

et qui est, par symétrie des tenseurs, équivalente à un système de 10 équations scalaires indépendantes [1]. Dans toute la suite :

- on ne pose pas $c = G = 1$ pour faciliter l'implémentation des équations en termes numériques
- on choisit la signature de la métrique $(-, +, +, +)$
- on utilise les indices grecs μ, ν pour les sommations sur les quatre dimensions d'espace-temps
- on utilise les indices romains i, j pour les sommations sur les trois dimensions d'espace
- les quantités indicées par 0 sont évaluées à l'instant présent
- les dérivées par rapport au temps cosmique sont notées par un point : $\dot{X} = \frac{\mathrm{d}X(t)}{\mathrm{d}t}$
- les dérivées par rapport au temps conforme η sont notées par un prime : $X' = \frac{\mathrm{d}X(\eta)}{\mathrm{d}\eta}$

1. Qui se ramènent à 6 équations indépendantes pour satisfaire les identités de Bianchi.

Le tenseur de Ricci intervenant dans l'équation 1.1 peut directement être calculé à partir de la métrique :

$$\boxed{R_{\mu\nu} = \partial_\rho \Gamma^\rho_{\nu\mu} - \partial_\nu \Gamma^\rho_{\rho\mu} + \Gamma^\rho_{\rho\lambda}\Gamma^\lambda_{\nu\mu} - \Gamma^\rho_{\nu\lambda}\Gamma^\lambda_{\rho\mu}} \tag{1.2}$$

où les $\Gamma^\alpha_{\beta\gamma}$ sont les connexions affines :

$$\boxed{\Gamma^\alpha_{\beta\gamma} = \frac{1}{2} g^{\alpha\delta}\left(\frac{\partial g_{\delta\beta}}{\partial x^\gamma} + \frac{\partial g_{\delta\gamma}}{\partial x^\beta} - \frac{\partial g_{\beta\gamma}}{\partial x^\delta}\right)} \tag{1.3}$$

si bien que $G_{\mu\nu}$ est entièrement déterminé par $g_{\mu\nu}$.

Cosmologie physique

L'équation de champ d'Einstein peut être vue comme le point de départ de la cosmologie physique car elle permet de calculer la dynamique de l'espace-temps à partir de son seul contenu énergétique. Toutefois, si l'on veut être en mesure de l'appliquer analytiquement à l'Univers dans son ensemble, et non pas seulement localement, il est nécessaire de lui adjoindre des symétries supplémentaires : c'est là qu'intervient le principe cosmologique. Ce principe stipule qu'à une échelle suffisamment grande, la distribution de matière dans l'Univers peut être considérée comme homogène et isotrope. D'où la construction classique de la cosmologie en un pendant homogène, respectant scrupuleusement le principe cosmologique, et un pendant perturbatif, consistant à appliquer de petites perturbations à la métrique homogène. Ce chapitre se limite à la première composante.

1.1.2 Équations de Friedmann-Lemaître

Forme de la métrique

La métrique de Friedmann-Lemaître-Robertson-Walker (FLRW), développée dans les années 1920 et 1930, correspond aux solutions de l'équation 1.1 pour un Univers satisfaisant le principe cosmologique. Sous les hypothèses d'homogénéité et d'isotropie, il est possible d'écrire la métrique :

$$\boxed{\mathrm{d}s^2 = -c^2\,\mathrm{d}t^2 + a\left(t\right)^2 \gamma_{ij}\,\mathrm{d}x^i\,\mathrm{d}x^j = -c^2\,\mathrm{d}t^2 + a\left(t\right)^2\,\mathrm{d}\Sigma^2} \tag{1.4}$$

avec en coordonnées hypersphériques :

$$\mathrm{d}\Sigma^2 = \mathrm{d}r^2 + f_k\left(r\right)\left(\mathrm{d}\theta^2 + \sin^2\theta\,\mathrm{d}\phi^2\right) \qquad \text{et} \qquad f_k\left(r\right) = \begin{cases} \frac{1}{\sqrt{|k|}}\sinh\left(r\sqrt{|k|}\right) & \text{si} \quad k < 0 \\ r & \text{si} \quad k = 0 \\ \frac{1}{\sqrt{|k|}}\sin\left(r\sqrt{|k|}\right) & \text{si} \quad k > 0 \end{cases} \tag{1.5}$$

où :

- $\mathrm{d}\Sigma$ fournit une mesure de la distance comobile, c'est-à-dire la distance qui sépare deux observateurs sans mouvement propre en faisant abstraction de l'expansion
- t est le temps cosmique, c'est-à-dire le temps propre de tous les observateurs au repos d'un Univers homogène et isotrope
- r, θ, ϕ correspondent aux coordonnées spatiales avec r proportionnel à la distance radiale
- $a\left(t\right)$ est le facteur d'échelle, sans dimension, que l'on fixe à $a_0 = 1$ aujourd'hui
- k, homogène à l'inverse d'une distance au carrée, caractérise la courbure (hyperbolique, euclidienne ou sphérique)

Solutions

Il devient alors possible de calculer la dynamique d'un Univers homogène et isotrope en résolvant l'équation de champ d'Einstein pour la métrique 1.4. En effet, en posant le paramètre de Hubble :

$$\boxed{H(t) = \frac{1}{a(t)}\frac{\mathrm{d}a(t)}{\mathrm{d}t}} \quad \text{et} \quad \begin{cases} H_0 = H(0) \\ h_0 = \dfrac{H_0}{100\,\mathrm{km/s/Mpc}} \end{cases} \tag{1.6}$$

et en calculant les connexions affines, puis le tenseur de Ricci, puis le scalaire de courbure, on obtient le tenseur d'Einstein $G_{\mu\nu}$ en fonction de H et de k. Les conditions de symétrie FLRW imposent au tenseur énergie-impulsion de correspondre à celui d'un fluide parfait :

$$\boxed{T_{\mu\nu} = \left(\rho c^2 + P\right) u_\mu u_\nu + P g_{\mu\nu}} \tag{1.7}$$

avec :

- u_μ la quadri-vitesse telle que $u_\mu u^\mu = -1$ c'est-à-dire $u_\mu = -c\delta^0_\mu$ et $u^\mu = \frac{1}{c}\delta^\mu_0$
- ρ la densité massique du fluide, lié à la densité d'énergie par $\rho_e = \rho c^2$
- P la pression du fluide

si bien que l'on obtient :

$$\begin{aligned} G_{00} &= 3H^2 + 3\frac{kc^2}{a^2} & T_{00} &= \rho c^4 \\ G_{0i} &= 0 & T_{0i} &= 0 \\ G_{ij} &= -\left(3\frac{H^2}{c^2} + 2\frac{\dot{H}}{c^2} + \frac{k}{a^2}\right) a^2 \gamma_{ij} & T_{ij} &= P a^2 \gamma_{ij} \end{aligned} \tag{1.8}$$

Cela permet d'établir les équations dites de Friedmann-Lemaître permettant de calculer l'évolution du paramètre d'échelle a en fonction du temps, présentées ci-après sous deux formes équivalentes couramment utilisées :

$$\boxed{\begin{aligned} 3H^2 + 3\frac{kc^2}{a^2} &= 8\pi G\rho \\ -3\frac{H^2}{c^2} - 2\frac{\dot{H}}{c^2} - \frac{k}{a^2} &= \frac{8\pi G}{c^4} P \end{aligned}} \quad \Leftrightarrow \quad \boxed{\begin{aligned} H^2 = \left(\frac{\dot{a}}{a}\right)^2 &= \frac{8\pi G}{3}\rho - \frac{kc^2}{a^2} \\ \dot{H} + H^2 = \frac{\ddot{a}}{a} &= -\frac{4\pi G}{3}\left(\rho + \frac{3P}{c^2}\right) \end{aligned}} \tag{1.9}$$

et à partir desquelles, on peut retrouver la conservation de l'énergie :

$$\boxed{\dot{\rho} = -3H\left(\rho + \frac{P}{c^2}\right)} \tag{1.10}$$

1.1.3 Contenu de l'Univers

Pour aller plus loin, il est nécessaire de paramétrer le contenu énergétique de l'Univers par une équation d'état w [2] :

$$\boxed{P = w\rho c^2} \quad \text{avec} \quad w = \begin{cases} 1/3 & \text{pour la radiation } R \\ 0 & \text{pour la matière } M \\ -1 & \text{pour la constante cosmologique } \Lambda \end{cases} \tag{1.11}$$

2. La nature de la constante cosmologique est traitée plus loin dans ce chapitre : à ce stade, on écrit juste l'équation d'état d'un fluide de pression négative.

C++11 code: RK4 integration of standard Friedmann-Lemaître equations

```cpp
/* Integration of standard FLRW cosmology equations
 * The following parameters are in SI units:
 * hubble_0: hubble constant
 * omega_r0: density parameter of radiation
 * omega_m0: density parameter of matter
 * omega_l0: density parameter of lambda
 * dt: integration time step
 * dt_backup: time step between backups
 * t_backward: time limit of backward integration
 * t_forward: time limit of forward integration
 * The function returns a container of pairs of t
 * and a corresponding to the cosmological evolution.
 */
template <typename T,
          class C = std::vector<std::pair<T, T>>,
          class = typename std::enable_if<
          std::is_floating_point<T>::value>::type>
C integrate_flrw(const T hubble_0,
                 const T omega_r0,
                 const T omega_m0,
                 const T omega_l0,
                 const T dt,
                 const T dt_backup,
                 const T t_backward,
                 const T t_forward)
{
  // Initialization
  static constexpr std::size_t off = 16;
  static constexpr T zero = 0;
  static constexpr T one = 1;
  static constexpr T two = 2;
  static constexpr T three = 3;
  static constexpr T six = 6;
  static constexpr T w_r = one/three;
  static constexpr T w_m = zero;
  static constexpr T w_k = -one/three;
  static constexpr T w_l = -one;
  static constexpr T a_0 = one;
  const T omega = omega_r0+omega_m0+omega_l0;
  const T omega_k0 = one-omega;
  C past = {std::make_pair(zero, a_0)};
  C future = {std::make_pair(zero, a_0)};

  // Expression of da/dt
  auto da_dt = [&](const T a, const T sign) {
    return sign*a*hubble_0*std::sqrt(
     omega_r0*std::pow(a, -three*(one+w_r))
    +omega_m0*std::pow(a, -three*(one+w_m))
    +omega_k0*std::pow(a, -three*(one+w_k))
    +omega_l0*std::pow(a, -three*(one+w_l)));
  };

  // Runge-kutta of order 4
  auto rk4 = [&](const T a, const T sign) {
    const T k1 = da_dt(a, sign);
    const T a1 = a+dt/two*k1;
    const T k2 = da_dt(a1, sign);
    const T a2 = a+dt/two*k2;
    const T k3 = da_dt(a2, sign);
    const T a3 = a+dt*k3;
    const T k4 = da_dt(a3, sign);
    return a+dt/six*(k1+k2+k2+k3+k3+k4);
  };

  // Integration loop
  auto loop = [&](C& c, const T sign, const T lim) {
    T t_sign = sign;
    T t_back = c.back().first;
    T t_old = t_back;
    T t_new = t_back;
    bool t_ok = std::abs(t_new) < std::abs(lim);
    T a_sign = sign;
    T a_back = c.back().second;
    T a_old = a_back;
    T a_new = a_back;
    bool a_ok = a_new > zero && std::isnormal(a_new);
    T origin = c.back().first;
    c.reserve(std::abs((t_back-lim)/dt_backup)+off);
    for (std::intmax_t i = 0; t_ok && a_ok; ++i) {
      t_new = origin+i*sign*dt;
      a_new = rk4(a_old, sign);
      a_ok = !(std::isnan(a_new) && a_old > a_0);
      if (!a_ok && !std::signbit(a_sign*sign)) {
        a_sign = -a_sign;
        a_new = rk4(a_old, a_sign);
      }
      t_ok = std::abs(t_new) < std::abs(lim);
      a_ok = a_new > zero && std::isnormal(a_new);
      if (t_ok && a_ok) {
        t_old = t_new;
        a_old = a_new;
        if (!(std::abs(t_back-t_new) < dt_backup)) {
          t_back = t_new;
          a_back = a_new;
          c.emplace_back(t_back, a_back);
        }
      } else if (std::abs(t_old-t_back) > zero) {
        c.emplace_back(t_old, a_old);
      }
    }
  };

  // Execution of backward and forward integration
  std::thread backward(loop, std::ref(past), -one,
                       t_backward);
  std::thread forward(loop, std::ref(future), one,
                      t_forward);

  // Finalization
  backward.join();
  forward.join();
  future.insert(future.begin(), past.rbegin(),
                past.rend()-1);
  return future;
}
```

Figure 1 – Exemple d'implémentation de résolution de l'équation 1.15. L'algorithme repose sur un simple intégrateur Runge-Kutta explicite d'ordre 4. L'intégration débute en $a = a_0$ et s'effectue simultanément vers le passé et vers le futur sur deux threads différents. La fonction retourne un conteneur temporaire de paires de valeurs t et a.

le contenu global pouvant alors s'exprimer comme un mélange pondéré de ces différents fluides[6]. La conservation de l'énergie 1.10 pour chaque fluide X donne :

$$\rho_X \propto a^{-3(1+w_X)} \tag{1.12}$$

Conjointement à ces considérations, la densité critique associée à un Univers de courbure nulle se déduit de la première équation de Friedmann-Lemaître :

$$\boxed{\rho_c = \frac{3H^2}{8\pi G}} \tag{1.13}$$

ce qui conduit à poser les paramètres de densité :

$$\boxed{\Omega_X = \frac{\rho_X}{\rho_c} = \frac{8\pi G\rho_X}{3H^2}} \quad \text{avec} \quad \begin{cases} \Omega = \sum_X \Omega_X \\ \Omega_k = 1 - \Omega \\ \Omega_\Lambda = \frac{\Lambda c^2}{3H^2} \end{cases} \tag{1.14}$$

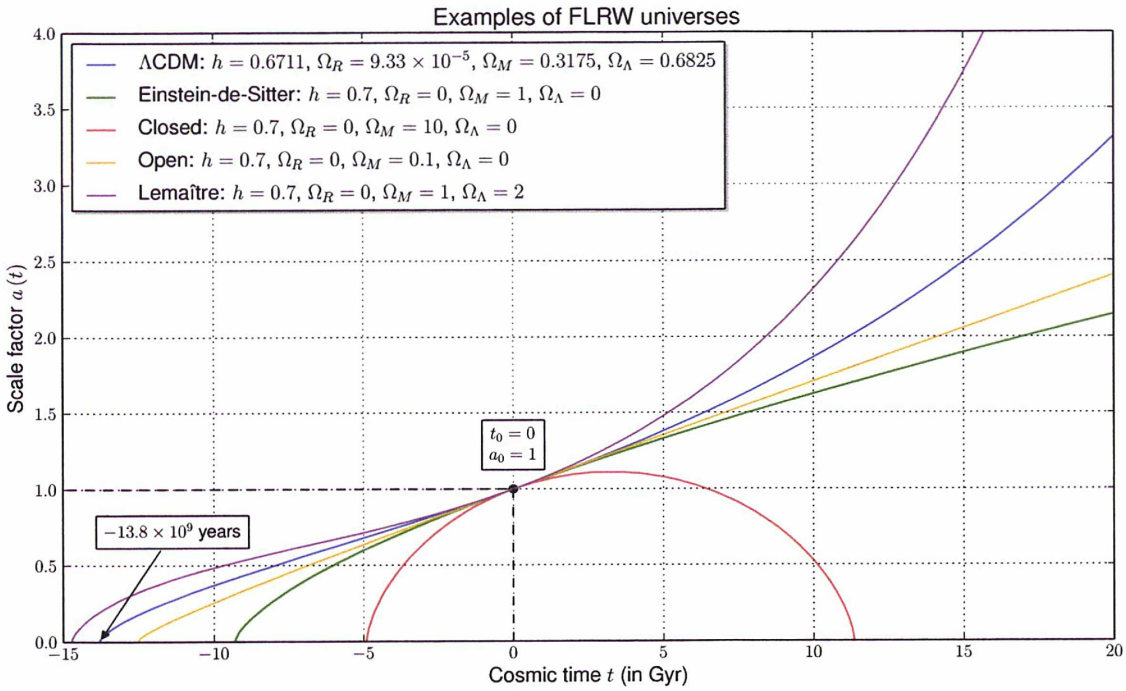


Figure 2 – Evolution de quelques solutions de modèles d'univers homogène et isotrope. Le modèle ΛCDM correspond au cas homogène déduit des analyses de la mission Planck[7] : la singularité se produit il y a environ 13.8 milliards d'années et l'expansion s'accélère peu à peu. Le cas Einstein-de-Sitter est celui d'un univers empli de matière et de courbure nulle : le taux d'expansion tend alors asymptotiquement vers zéro. Pour les cas fermé et ouvert, la courbure est non-nulle : après une phase d'expansion, le second connaît une phase de contraction. L'équation 1.15 et l'intégrateur de la figure 1 permettent ainsi d'explorer une vaste gamme de modèles cosmologiques.

si bien que 1.9 peut se réécrire :

$$\boxed{\frac{H^2}{H_0^2} = \Omega_{R_0} a^{-4} + \Omega_{M_0} a^{-3} + \Omega_{k_0} a^{-2} + \Omega_{\Lambda_0}} \tag{1.15}$$

Cette équation différentielle sur a constitue un point nodal de la cosmologie homogène car elle permet d'obtenir les solutions des modèles FLRW standard à partir d'une simple paramétrisation en termes de H_0, Ω_{R_0}, Ω_{M_0} et Ω_{Λ_0}. Grâce à elle, il est donc possible d'explorer une vaste gamme de modèles, certains exhibant des comportements d'expansion accélérée, d'autres des phases de recontraction comme illustré en figure , obtenue à partir de l'intégrateur présenté en figure 1.

1.1.4 Quantités cosmologiques

Coordonnées comobiles et temps conforme

Une fois la cosmologie posée, un certain nombre de quantités physiques peut être calculé. Tout d'abord, d'autres coordonnées sont couramment utilisées pour la métrique friedmannienne 1.4. En effet, dans le cas d'une courbure nulle, il est d'usage dans les simulations cosmologiques de se placer en coordonnées comobiles cartésiennes :

$$\mathrm{d}\Sigma^2 = \mathrm{d}x^2 + \mathrm{d}y^2 + \mathrm{d}z^2 \tag{1.16}$$

et d'utiliser le temps conforme η :

$$\mathrm{d}\eta = \frac{\mathrm{d}t}{a} \qquad \Rightarrow \qquad \boxed{\eta = \int_0^t \frac{\mathrm{d}t'}{a\,(t')}} \tag{1.17}$$

ce qui permet de réécrire 1.4 sous la forme :

$$\mathrm{d}s^2 = a^2\left(-c^2\,\mathrm{d}\eta^2 + \mathrm{d}x^2 + \mathrm{d}y^2 + \mathrm{d}z^2\right) \tag{1.18}$$

Redshift

Au niveau des observables, la quantité centrale en cosmologie est le redshift z :

$$\boxed{1+z = \frac{\nu_S}{\nu_O} = \frac{(k_\nu u^\nu)_S}{(k_\nu u^\nu)_O} = \frac{(g_{\mu\nu}k^\mu u^\nu)_S}{(g_{\mu\nu}k^\mu u^\nu)_O}} \quad \underset{\text{FLRW}}{\Rightarrow} \quad z = \frac{a_0}{a} - 1 \quad \text{et} \quad a = \frac{a_0}{1+z} \tag{1.19}$$

ou décalage spectral qui correspond au rapport des fréquences lumineuses ν entre une source S et un observateur O. Dans cette expression, k^μ désigne les dérivées des coordonnées par rapport au paramètre affine et u^ν représente les quadri-vitesses de la source et de l'observateur. Par cette équation, la mesure du décalage des raies spectrales des galaxies fournit une évaluation directe du facteur d'échelle a à condition de supposer la cosmologie homogène et l'absence de vitesse péculière. Toute la difficulté de la mesure de $a\,(t)$ vient de la détermination de la distance exacte à laquelle se trouvent ces objets lointains.

Distance de Hubble

Dans toute la suite, avec $a_0 = 1$ on définit le paramètre de Hubble adimensionné[4] à partir de l'équation 1.15 :

$$E\,(z) = \sqrt{\Omega_{R_0}\,(1+z)^4 + \Omega_{M_0}\,(1+z)^3 + \Omega_{k_0}\,(1+z)^2 + \Omega_{\Lambda_0}} \tag{1.20}$$

D'autre part on définit le temps de Hubble t_H correspondant à l'âge de l'Univers pour un taux d'expansion constant et la distance de Hubble d_H comme :

$$t_H = \frac{1}{H_0} \qquad \text{et} \qquad d_H = ct_H = \frac{c}{H_0} \tag{1.21}$$

Distance de voyage de la lumière

Le « temps de regard vers le passé » ou lookback time est défini comme la différence entre l'âge actuel de l'Univers t_0 et l'âge de l'Univers au moment de l'émission de photons t_e :

$$t_T\,(z) = t_0 - t_e = t_H \int_0^z \frac{dz'}{(1+z')\,E\,(z')} \quad \text{et} \quad d_T\,(z) = ct_T\,(z) = d_H \int_0^z \frac{dz'}{(1+z')\,E\,(z')} \tag{1.22}$$

avec d_T la distance de voyage de la lumière.

Distance comobile

Pour un univers FLRW, la distance comobile sur la ligne de visée notée χ ou d_C à un redshift z est donnée par[8] :

$$\boxed{\chi = \int_{\eta_e}^{\eta_0} c\,\mathrm{d}\eta' = \int_{t_e}^{t_0} c\frac{\mathrm{d}t'}{a\,(t')}} \quad \Rightarrow \quad d_C\,(z) = d_H \int_0^z \frac{dz'}{E\,(z')} \tag{1.23}$$

tandis que la distance comobile transverse d_M est égale à :

$$d_M\,(z) = \begin{cases} \frac{d_H}{\sqrt{|\Omega_k|}} \sinh\left(\sqrt{|\Omega_k|}\frac{d_C(z)}{d_H}\right) & \text{si} \quad \Omega_k > 0 \\ d_C\,(z) & \text{si} \quad \Omega_k = 0 \\ \frac{d_H}{\sqrt{|\Omega_k|}} \sin\left(\sqrt{|\Omega_k|}\frac{d_C(z)}{d_H}\right) & \text{si} \quad \Omega_k < 0 \end{cases} \tag{1.24}$$

Distance de diamètre angulaire

La distance de diamètre angulaire reflète la distance apparente d'un objet cosmologique à partir de son diamètre angulaire :

$$\boxed{d_A = \frac{x}{\theta}} \quad \Rightarrow \quad d_A(z) = \frac{d_M(z)}{(1+z)} \tag{1.25}$$

où x est la taille physique de l'objet et θ l'angle sous lequel il est vu par l'observateur.

Distance de luminosité

Enfin, la distance de luminosité se déduit de la « dilution » du flux lumineux :

$$\boxed{d_L = \sqrt{\frac{L}{4\pi F}}} \tag{1.26}$$

avec L la luminosité intrinsèque de l'objet et F le flux lumineux mesuré par l'observateur. La relation de dualité d'Etherington[9] qui lie la distance de luminosité à la distance de diamètre angulaire :

$$\boxed{d_L = (1+z)^2 d_A} \tag{1.27}$$

reste valide pour toute cosmologie[10], y compris dans les cas inhomogènes.

1.2 Modèle de concordance ΛCDM

1.2.1 Élaboration du modèle

Toutes ces considérations issues de la relativité générale et des symétries découlant du principe cosmologique, et dont une grande partie a été élaborée dans les années 1920, restent toutefois cantonnées au domaine de la théorie. Et si la relativité générale est vérifiée, jusqu'à présent, au niveau local, la question de la confirmation ou de l'infirmation de ces modèles par l'observation restait ouverte. Car tous ces modèles décrivent un univers en expansion et/ou en contraction, ce qui, à une époque où la question de la nature des galaxies n'était pas encore tranchée [3] représentait tout de même un tournant philosophique. C'est d'ailleurs pour cette raison qu'Einstein avait introduit dès 1917 une constante, dite constante cosmologique Λ, dans ses équations permettant de rendre statique ces modèles :

$$R_{\mu\nu} - \frac{1}{2} g_{\mu\nu} R - \Lambda g_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu} \tag{1.28}$$

Il s'agit d'ailleurs du seul degré de liberté que permet la relativité générale originale. Vu sous un autre angle, l'équation 1.1 n'est que le cas spécifique $\Lambda = 0$ de l'équation 1.28 qui est une forme plus générale qui peut se déduire du principe d'équivalence fort. Il est toutefois nécessaire d'insister sur le fait que, dans ce cas, Λ est une constante de nature purement géométrique [4].

La confirmation de l'aspect dynamique de l'Univers viendra avec les observations d'Edwin Hubble en 1929 et la formulation de la loi qui porte aujourd'hui son nom, à savoir qu'il existe une relation linéaire entre le redshift des galaxies et leur distance relative établie par l'observation d'étoiles variables de type céphéide. Ce qui peut s'interpréter comme le fait que les galaxies s'éloignent les unes des autres, et ce d'autant plus rapidement que la distance qui les sépare est grande : autrement dit, si

3. Voir « le grand débat » des années 1920 au sujet de la nature des nébuleuses spirales entre Harlow Shapley et Heber Curtis.
4. En effet, Λ peut être intégré dans $T_{\mu\nu}$ auquel cas la constante se réinterprète comme une nouvelle composante énergétique et son caractère constant peut être relaxé, ce qui est par exemple le cas des modèles de quintessence.

nous ne sommes pas en un point très particulier de l'espace alors c'est que partout l'espace entre les galaxies augmente, c'est-à-dire que l'Univers lui-même est en expansion. Toutefois, pris isolément, cette observation n'implique pas nécessairement l'expansion [5]. Mais cette interprétation s'est imposée naturellement car elle venait s'inscrire dans une construction logique plus vaste, le modèle du Big-Bang, qui donnait de la cohérence à de multiples mesures et qu'il devenait de plus en plus difficile de contester à mesure que de nouvelles analyses devenaient possibles du fait de l'amélioration des techniques observationnelles. Aujourd'hui ce scénario basé sur la dynamique des métriques FLRW est corroboré par plusieurs éléments de natures différentes :

- le décalage vers le rouge décrit précédemment
- l'observation et la caractérisation du fond diffus cosmologique (surface de dernière diffusion) depuis sa découverte en 1964 par Penzias et Wilson
- l'abondance des éléments primordiaux (H, He, Li, D)
- l'évolution, la distribution et la morphologie des galaxies et des grandes structures
- l'enrichissement en métaux [6] des populations stellaires successives

Ainsi, si les univers en expansion découlant des métriques FLRW s'accordent plutôt bien avec les observations, la question la plus naturelle qui s'en suit est celle des paramètres du modèle. Autrement dit, celle de la valeur de la constante de Hubble H_0, de la courbure de l'Univers Ω_{k_0}, de ses constituants énergétiques et de leur proportion relative. Non parce que ces quantités ont une importance capitale intrinsèque – il ne s'agit pas de constantes fondamentales de la Nature – mais parce qu'elles permettent de tester la pertinence du modèle lui-même. Et c'est la détermination de ces paramètres qui a conduit progressivement à l'établissement du modèle de concordance ΛCDM qui prévaut aujourd'hui (voir figure 3) : celui d'un univers homogène et isotrope à grande échelle dont l'expansion est décrite par les équations de Friedmann-Lemaître et qui contient, en plus du rayonnement et de la matière baryonique classique, de la matière noire froide (CDM pour « Cold Dark Matter ») et une constante cosmologique Λ.

1.2.2 Matière noire

Le premier élément « non-standard » de ce modèle, est la matière noire, et plus spécifiquement la matière noire froide, c'est-à-dire non relativiste. Cette matière noire tente de répondre à un vieux problème qui avait été soulevé dès les années 1930 par Zwicky et Smith, et qui avait été relayé jusqu'aux années 1970 au statut de problème annexe au profit de l'élaboration et de la validation des modèles cosmologiques. Ce problème c'est celui des courbes de rotation des galaxies spirales et de la dynamique des amas galactiques. Il ne s'agissait donc pas, à l'origine, d'un problème de nature cosmologique, mais plutôt d'un problème de dynamique galactique.

Ce problème provient du désaccord existant entre la masse dynamique et la masse lumineuse des galaxies spirales vues par la tranche. En effet, la vitesse de rotation ω de ces objets astrophysiques peut être calculée de deux façons indépendantes :

- ω_D, par mesure directe des décalages spectraux à différents endroits dans le disque galactique
- ω_L, par le calcul newtonien basé sur la masse lumineuse en différents points du disque, en supposant que celui-ci est exclusivement formé d'étoiles

Or, comme l'illustre l'exemple de la figure [?], les résultats issus de ces deux méthodes de mesure diffèrent systématiquement : à mesure que l'on s'éloigne du centre galactique, ω_D se maintient sur un plateau tandis que ω_L décroit. Comme toujours en physique, cette anomalie peut trouver son origine dans des explications de trois natures différentes : soit le cadre théorique n'est plus valide à de si grandes

5. A titre d'exemple, la théorie de la « lumière fatiguée » de Zwicky en est une autre.
6. En astrophysique, tous les éléments plus lourds que l'hydrogène et l'hélium.

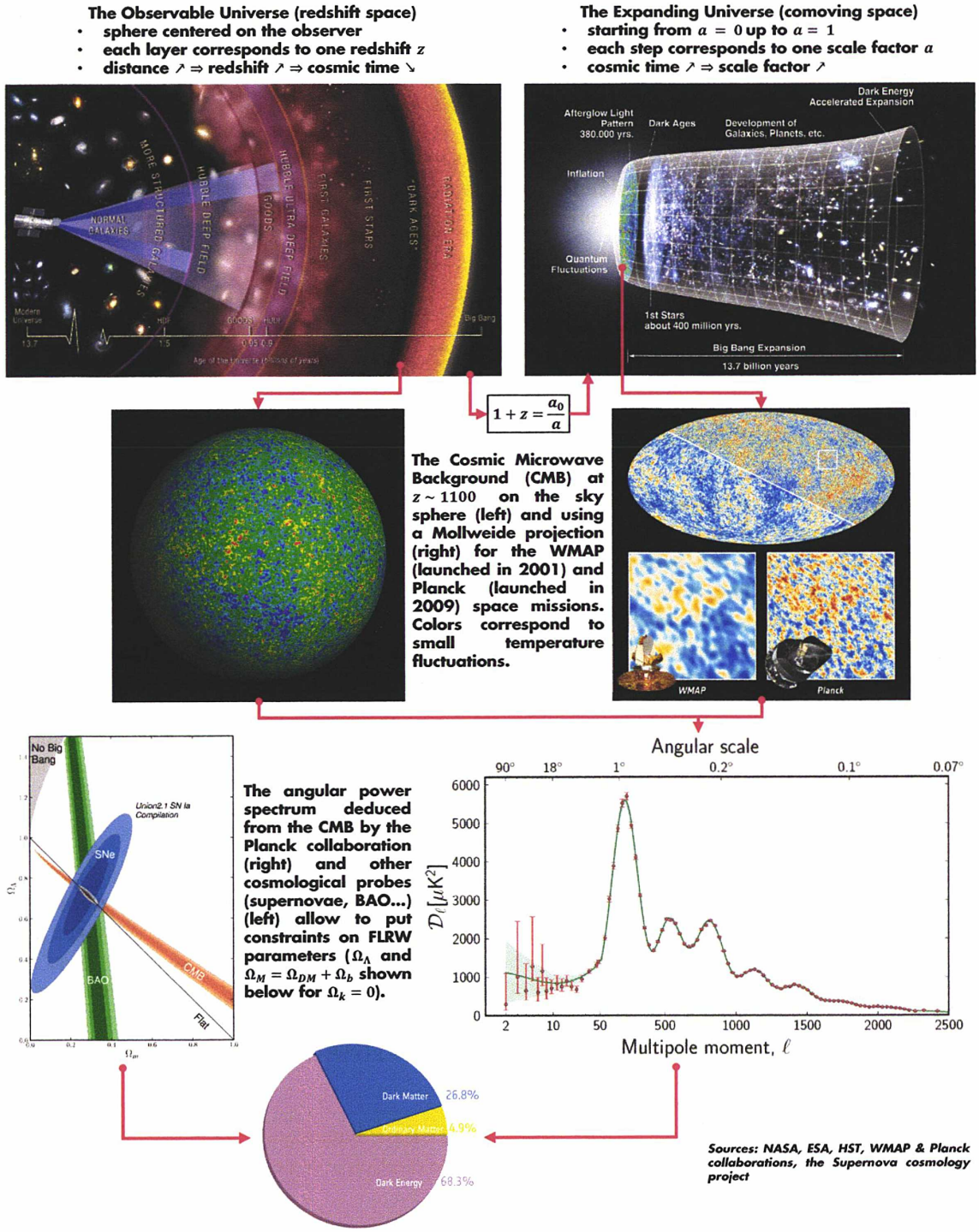


FIGURE 3 – Illustration de quelques éléments de construction du modèle de concordance. L'univers en expansion peut être envisagé dans le cadre de deux représentations complémentaires : l'espace des redshifts d'une part, correspondant à l'Univers Observable tel que nous le voyons, et l'espace comobile d'autre part, dont chaque « tranche » est prise à un temps cosmique donné. La correspondance de l'un à l'autre passe par l'association d'un redshift z à un facteur d'échelle a. Au sein de cet univers, une partie des contraintes sur les paramètres cosmologiques vient de l'observation du fond diffus cosmologique (CMB) datant de l'époque de la recombinaison, et notamment du spectre de puissance des fluctuations de température à cette époque, ici mesuré par la mission PLANCK. Dans le modèle de concordance, ce sont ces infimes variations de $\frac{dT}{T} \sim 1 \times 10^{-5}$, qui vont par la suite engendrer la formation de structures de plus en plus grandes pour aboutir aux amas et aux superamas de galaxies que nous connaissons aujourd'hui. Combiné à d'autres sondes (supernovæ, oscillations acoustiques des baryons (BAO) ...), le CMB de PLANCK donne des valeurs de $\Omega_{\Lambda_0} \approx 0.683$, $\Omega_{M_0} = \Omega_{DM_0} + \Omega_{b_0} \approx 0.268 + 0.049$ et $H_0 \approx 68.7\,\mathrm{km/s/Mpc}$ pour les principaux paramètres cosmologiques du modèle de concordance.

échelles et il est nécessaire d'amender la relativité générale ; soit les hypothèses de calcul sont abusives et/ou les données sont mal interprétées ; soit enfin les lois sont justes, les données sont correctement interprétées et il existe par conséquent un élément non observé dont cette différence est la manifestation. Ce dernier cas correspond justement à ce que l'on appelle matière noire. Le raisonnement est simple : s'il existe un tel écart entre ω_L et ω_D, c'est que la galaxie n'est pas seulement constituée d'étoiles, mais aussi d'éléments qui ne rayonnent pas et dont la masse suffit à rehausser ω_L pour l'amener au niveau du plateau observé par la mesure des décalages spectraux.

Dès lors, chaque galaxie n'est plus seulement caractérisée par sa masse visible constituée par les étoiles et les nuages de gaz, mais également par le halo de matière noire au sein duquel elle s'est formée. A l'heure actuelle, l'hypothèse de la matière noire n'est plus seulement motivée par la dynamique des galaxies spirales. En effet, le spectre de puissance des anisotropies du fond diffus cosmologique, les effets de lentille gravitationnelle faible au sein des amas et des superamas de galaxies, et les oscillations acoustiques de baryons sont autant d'éléments favorables à un paramètre de densité de la matière noire Ω_{DM_0} environ cinq fois supérieur à celui de la matière baryonique Ω_{b_0}. Ou disons plutôt que supposer la présence de cette matière est une hypothèse minimale qui permet de retrouver toutes ces mesures observationnelles indépendantes avec une bonne précision.

Le problème qui demeure est évidemment celui de la nature fondamentale de cette matière non lumineuse. Et pour le moment, les recherches de WIMPs [7] se sont avérées infructueuses[11], ou tout du moins les expériences n'ont pas accumulé une statistique suffisante pour apporter une conclusion à ces recherches. Mais, les propriétés de la matière noire peuvent aussi être inférées à partir des observations astronomiques. Et des évènements rares comme celui du Bullet Cluster[12] ont déjà permis de valider la nature non collisionnelle de la matière noire si telle est bien la cause de l'écart entre la structure observée de l'amas et le potentiel gravitationnel inféré à partir de l'effet de lentillage qu'il produit. Dans tous les cas, il s'agit là d'une pierre angulaire du modèle de concordance qui permet de retrouver un bon accord avec les observations en limitant le nombre d'hypothèses et leur nature.

1.2.3 Énergie noire

Le second élément « non-standard » de ΛCDM, c'est bien sûr l'énergie noire. Elle est rendue nécessaire dans ΛCDM pour modéliser l' accélération de l'expansion mise en évidence en 1998 par la High-Z Supernova Search Team[13] puis en 1999 par le Supernova Cosmology Project[14] comme illustré en figure 5. Cette mise en évidence procède de la mesure de la distance luminosité en fonction du redshift $D_L(z)$ obtenue à partir de l'observation de supernovæ de type Ia dont la magnitude absolue peut être déterminée via l'évolution de la magnitude apparente comme illustré en figure 4. Comme pour le problème des courbes de rotation des galaxies spirales, cette anomalie observationnelle peut être expliquée par trois approches : l'une législative, consistant à étendre ou à modifier la relativité générale, l'une ontologique, consistant à ajouter une nouvelle forme d'énergie au modèle, et l'une paradigmatique consistant à s'interroger sur la validité des hypothèses du modèle et notamment sur le bien-fondé du principe cosmologique.

Mais depuis l'observation historique des supernovæ, tout un faisceau de preuves s'est accumulé en faveur de la nécessité d'une nouvelle composante énergétique. Le fond diffus cosmologique est compatible avec un Univers plat $\Omega_k = 0$ alors que la matière, baryonique et noire, ne représente qu'environ 30% de la densité critique. D'autre part, les mesures d'effet Sachs-Wolfe intégré[15, 16] témoignent également d'une accélération de l'expansion, bien que l'amplitude de l'effet soit encore débattu. Or, comme dans le cas de la matière noire, l'introduction de cette seule constante cosmologique permet de retrouver un très bon accord entre théorie et observations. Sauf que là encore, la nature exacte de cette composante est aujourd'hui complètement inconnue.

7. Weakly interacting massive particles.

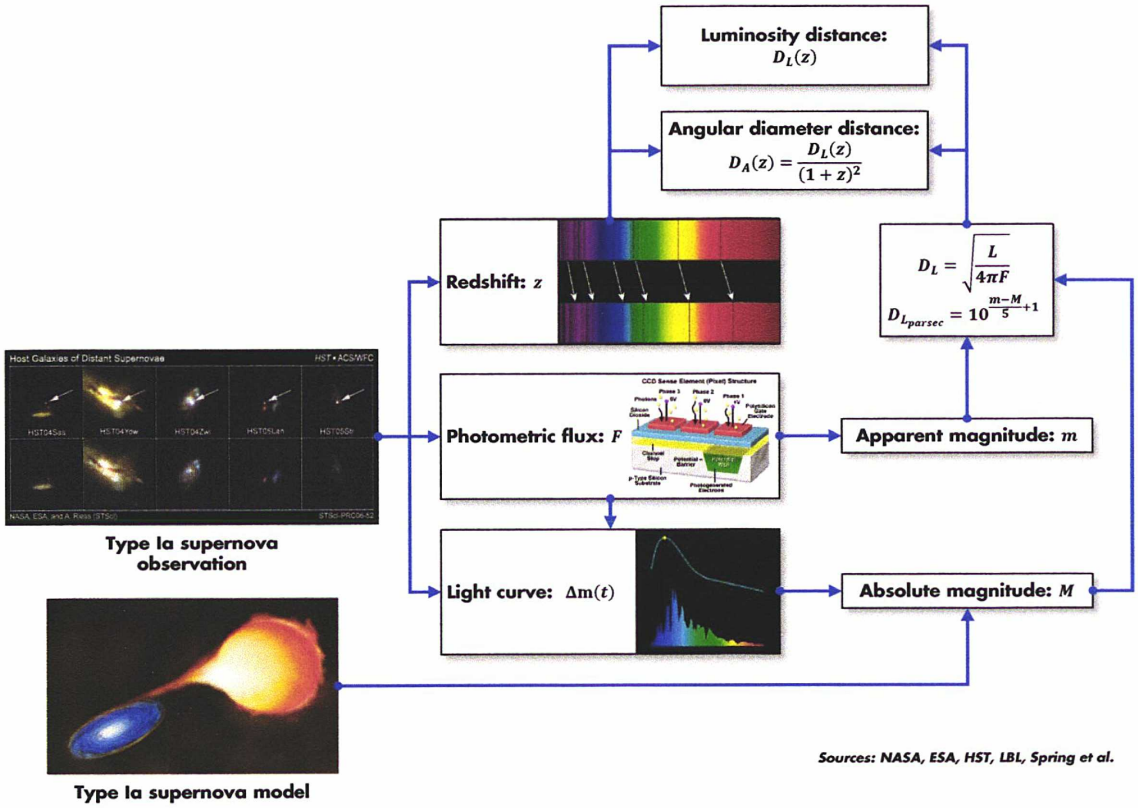


FIGURE 4 – Principe de détermination de la distance de luminosité et de la distance de diamètre angulaire en fonction du redshift à partir d'observations de supernovae de type Ia. L'observation d'une supernova de type Ia permet de mesurer directement deux quantités : son redshift z et son flux lumineux F. De ce dernier en est déduite la magnitude apparente m de l'objet. De plus, l'observation dans la durée de la supernova permet d'obtenir sa variation de luminosité en fonction du temps, de laquelle peut se déduire sa magnitude absolue M. La distance luminosité D_L est directement calculable à partir de m et M de même que la distance de diamètre angulaire D_A par la relation d'Etherington 1.27. En répétant l'opération sur un grand nombre de supernovæ, il est donc possible d'obtenir finalement la dépendance de D_L et de D_A en fonction du redshift. Il est important de noter qu'aucun modèle cosmologique n'intervient dans ce raisonnement : la cosmologie n'intervient qu'à partir du moment où l'on souhaite déduire les paramètres cosmologiques de $D_L\,(z)$.

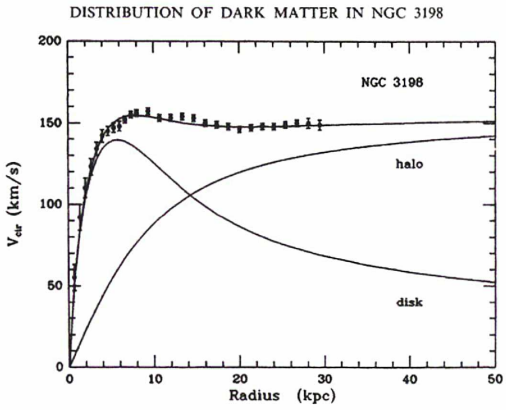
Comme précisé à l'occasion de la présentation de l'équation 1.28, il est toutefois nécessaire de distinguer constante cosmologique et énergie noire. En effet, même si ces dernières, dans le cadre de ΛCDM ont des effets strictement équivalent, l'interprétation physique qui en est fait est sensiblement différente. Dans le premier cas, celui d'une constante cosmologique Λ, il s'agit d'un terme purement géométrique intervenant comme une modification, ou plutôt une généralisation, de l'équation de champ d'Einstein : il s'agit donc là d'une approche législative au problème de l'accélération de l'expansion. Dans le second cas, celui d'une approche ontologique, l'équation de champ d'Einstein est conservée sous sa forme 1.1, et une nouvelle composante est incorporée au tenseur énergie-impulsion :

$$T_{\mu\nu}^{(\Lambda)} = \frac{c^4\Lambda}{8\pi G} g_{\mu\nu} \tag{1.29}$$

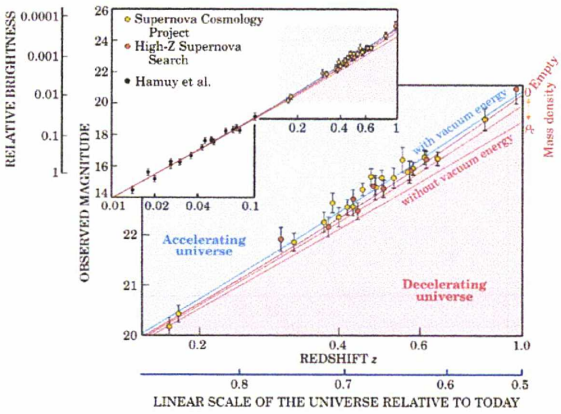
Cette énergie noire d'équation d'état w_Λ se comporte ainsi comme un fluide de pression négative :

$$\boxed{w_\Lambda = -1} \qquad \begin{cases} \rho_\Lambda = \frac{c^2\Lambda}{8\pi G} \\ P_\Lambda = -\frac{c^4\Lambda}{8\pi G} \end{cases} \tag{1.30}$$

Alors que dans le premier cas, il ne peut s'agir que d'une constante, le second ouvre les portes de modélisations plus fines, avec par exemple des équations d'état variables au cours du temps.



(a) Courbe de rotation de NGC 3198[17].



(b) Magnitude de SNIa en fonction du redshift[13, 14].

Figure 5 – Exemples de motivations observationnelles pour la matière et l'énergie noires. A gauche, la confrontation des mesures directes de la vitesse de rotation de la galaxie NGC 3198 par décalage spectral (points de données et barres d'erreurs) à la vitesse de rotation obtenue par l'estimation de la masse du disque galactique soulève le problème de la matière noire. En effet, alors que la dynamique du disque devrait conduire à une diminution de la vitesse de rotation au délà de 6 kpc, celle-ci se maintient sur un plateau. La présence d'un halo de matière noire permet de réconcilier les deux. A droite, la mesure de la magnitude apparente et de la distance luminosité de supernovæ de type Ia situés à différents redshifts par deux expériences différentes se révèle être incompatible avec une métrique FLRW sans constante cosmologique. L'écart observé témoigne d'une accélération de l'expansion, modélisable par une énergie noire.

ΛCDM cosmological parameters				
	H_0 [km/s/Mpc]	Ω_{R_0}	Ω_{M_0}	Ω_{Λ_0}
WMAP7+BAO+H_0	70.4	8.41×10^{-5}	0.272	0.728
Planck2013+WP	67.04	9.351×10^{-5}	0.3183	0.6817

Figure 6 – Best fit des paramètres cosmologiques principaux du modèle de concordance pour WMAP7 et Planck. La première ligne correspond aux valeurs obtenues après sept ans d'observations par la mission WMAP en tenant compte de la mesure des oscillations acoustiques de baryons et de mesures locales de la constante de Hubble[18]. La seconde correspond aux résultats de la mission Planck publiés en 2013[7] en tenant compte de la polarisation mesurée par WMAP à bas l. La valeur de Ω_{R_0} est déduite de la relation $\Omega_{R_0}\,(1 + z_{eq}) = \Omega_{M_0}$ où z_{eq} est le redshift d'équivalence rayonnement-matière (voir figure 7). Dans tous les cas il s'agit de « best fits » : les procédures de réduction et les barres d'erreurs sur les paramètres cosmologiques sont détaillées dans les articles des deux programmes d'observation.

1.2.4 Propriétés dynamiques

Paramètres cosmologiques

Sous ces hypothèses d'isotropie et d'homogénéité à grande échelle, de présence de matière noire froide et d'énergie noire, le modèle de concordance permet de retrouver un très bon accord avec les observations cosmologiques. La valeur des paramètres intervenant dans l'équation 1.15 donnée par les missions WMAP et Planck est présentée en figure 6. Dans toute la suite de ce chapitre, on considère les valeurs données par Planck.

Univers jeune

Connaissant H_0, Ω_{R_0}, Ω_{M_0} et Ω_{Λ_0}, il est alors possible de construire un certain nombre de repères qui seront utiles dans toute la suite. Tout d'abord ΛCDM est caractérisé par trois ères, correspondant à la prédominance d'une composante énergétique sur les autres comme illustré en figure 7 : une ère primordiale de rayonnement, une ère de matière, et plus récemment, une ère de constante cosmologique, caractérisée par une accélération de l'expansion de plus en plus prononcée. Ces trois phases

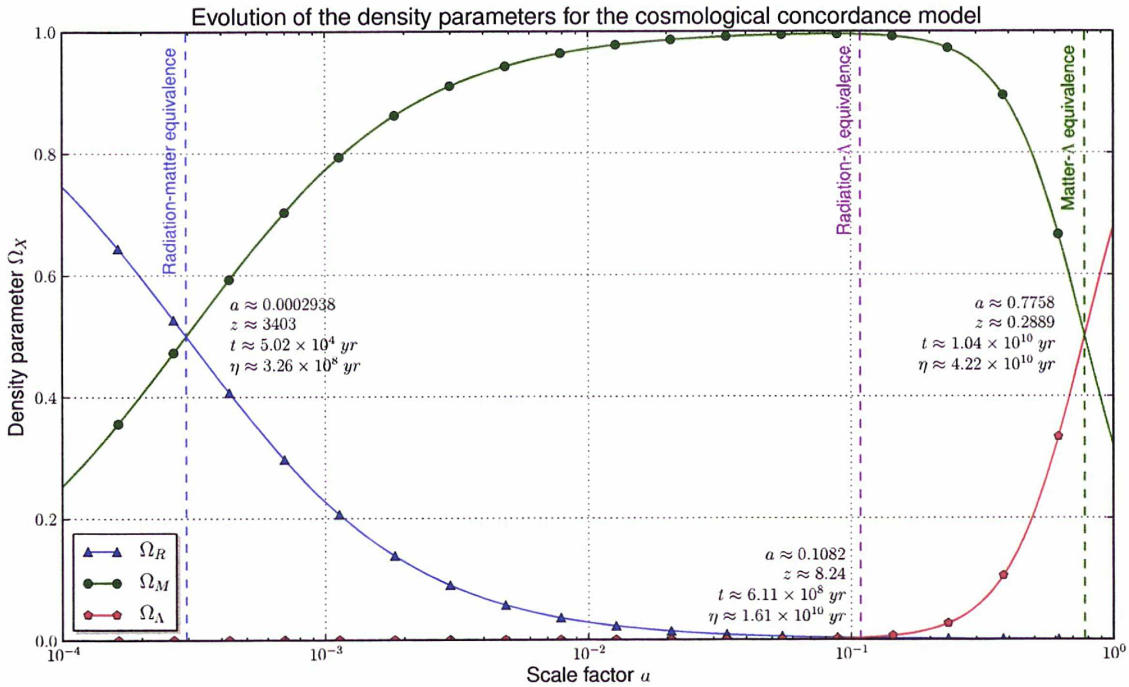


Figure 7 – Ères de rayonnement, de matière et de constante cosmologique. L'évolution du facteur d'échelle est liée à la dynamique de trois ères successives correspondant à la dominance du paramètre de densité d'un type particulier de composante sur les autres. Ces trois ères trouvent leur origine dans le fait que $\rho_R(a) = \rho_{R_0} a^{-4}$, $\rho_M(a) = \rho_{M_0} a^{-3}$ et $\rho_\Lambda(a) = \rho_{\Lambda_0}$ avec $\rho(a) = \rho_R(a) + \rho_M(a) + \rho_\Lambda(a)$. Dans la phase primordiale de l'Univers et jusqu'au redshift d'équivalence rayonnement-matière z_{eq}, le mix énergétique est largement dominé par le rayonnement. Puis, à mesure que le facteur d'échelle augmente, l'Univers entre dans une longue phase dominée par la matière. La recombinaison se produit à $a \approx 0.92 \times 10^{-3}$: l'hydrogène se forme et le rayonnement se découple de la matière. Le fond diffus cosmologique correspond à la surface de dernière diffusion datant de cette époque. Ce n'est que récemment, à un redshift légèrement inférieur à $z = 0.3$ que l'Univers est entré dans une ère dominée par la constante cosmologique, responsable d'un comportement accéléré.

d'expansion fournissent une grille de lecture à l'histoire de la dynamique de l'Univers selon ΛCDM présentée en figure 8. Le « best fit » des paramètres de Planck correspond à un âge de l'Univers d'environ 13.824 milliards d'années. En $z_{\text{CMB}} \approx 1090$, c'est-à-dire environ 370 000 ans après la singularité, se produit la recombinaison e^-/p^+ qui entraîne le découplage rayonnement/matière : l'Univers devient transparent et gardera la trace des fluctuations de température $\frac{dT}{T} \sim 10^{-5}$ à cet instant au travers du CMB. En tant que surface de dernière diffusion, ce dernier fixe ainsi la limite de la cosmologie observationnelle non gravitationnelle. Et c'est justement à partir de l'analyse de ces anisotropies illustrées en figure 2 que les paramètres cosmologiques sont mesurés par des missions comme Planck. En plus de fournir des éléments sur l'état de l'Univers à cette époque, le CMB garde les traces des mécanismes inflationnaires s'étant produits entre $t \sim 10^{-36}$ s et $t \sim 10^{-32}$ s, si inflation il y a bien eu. La détection et l'analyse de tels effets, et notamment la polarisation des modes B, pose cependant encore un certain nombre de difficultés[19].

Univers observable

Entre $z = z_{\text{CMB}}$ et $z = 0$ s'étend donc le domaine de la cosmologie observationnelle. D'un point de vue purement homogène, c'est-à-dire exception faite de toutes les problématiques de formation des structures, cette phase d'expansion peut être résumé en quatre redshifts caractéristiques, également représentés en figure 8 :

- la réionisation qui débute vers $z \approx 11.3$ et qui est liée au rayonnemment de la première génération d'étoiles

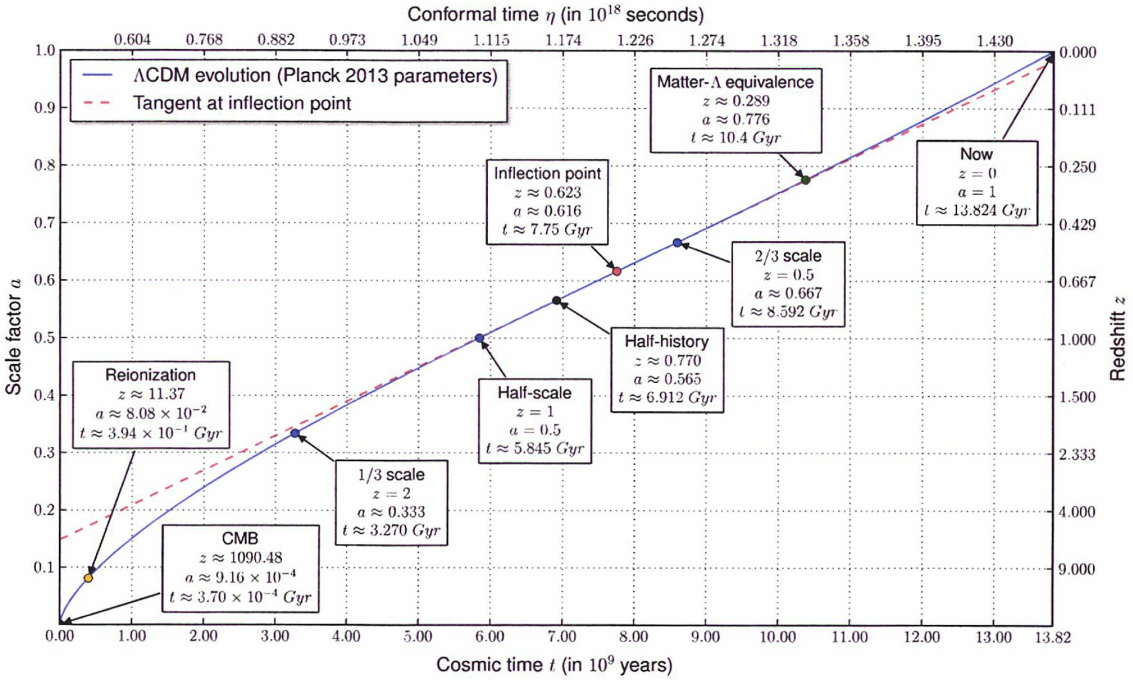


FIGURE 8 – Évolution du facteur d'échelle et repères chronologiques du modèle de concordance. La résolution de l'équation 1.15 pour les paramètres cosmologiques donnés par Planck permet de tracer l'évolution du paramètre d'échelle en fonction du temps cosmique t, c'est-à-dire la représentation de l'évolution des distances au cours de l'histoire d'un univers homogène et isotrope compatible avec les observations. Un certain nombre de repères peuvent alors être construits, en commençant par la correspondance entre t, η, a et z présentés sur les axes de la figure. Le point d'inflexion associé à $\ddot{a}(t) = 0$ marque la transition entre une phase de décélération et la phase actuelle d'accélération de l'expansion. La mission EUCLID prévoit de reconstruire la distribution de matière jusqu'à $z \approx 2$. A titre d'information, le plus ancien quasar connu à l'heure actuelle est situé à $z \approx 7.1$[20] et les plus anciennes galaxies ont été détectées légèrement au-delà de $z = 10$[21].

- l'Univers à la moitié de son histoire actuelle à $z \approx 0.77$
- le point d'inflexion à $z \approx 0.62$ correspondant au passage d'une phase de décélération à une phase d'accélération de l'expansion
- l'équivalence matière/constante cosmologique à $z \approx 0.29$

Problématique de la mesure des distances

Au sein de ce fond homogène, la matière va naturellement s'effondrer pour former des structures de plus en plus grandes. L'observation et la caractérisation de ces structures, qui représente un pan entier de la cosmologie observationnelle, passe par la détermination de la distance à laquelle elles se trouvent. La question qui se pose est alors la suivante : à partir de l'évaluation de la distance de ces objets, comment remonter à la cosmologie sous-jacente ? Et dans ce cadre, la distance de luminosité et la distance de diamètre angulaire ont un rôle central à jouer. D'un point de vue observationnel, il va s'agir de les déterminer indépendamment de la cosmologie, comme déjà illustré en figure 4. D'un point de vue théorique, l'objectif va être de les calculer dans le cadre d'un ensemble de paramètres cosmologique donné. Pour les paramètres de PLANCK, elles sont présentées en figure 9. Mais si dans le cas homogène, la correspondance entre la voie observationnelle et théorique est facile à établir, elle devient beaucoup plus difficile à formuler dès lors que l'on se pose la question de l'impact des grandes structures sur la mesure des distances elle-même. Or comme de cette correspondance dépend la détermination des paramètres cosmologiques, il s'agit d'un problème de première importance. Et c'est justement cette

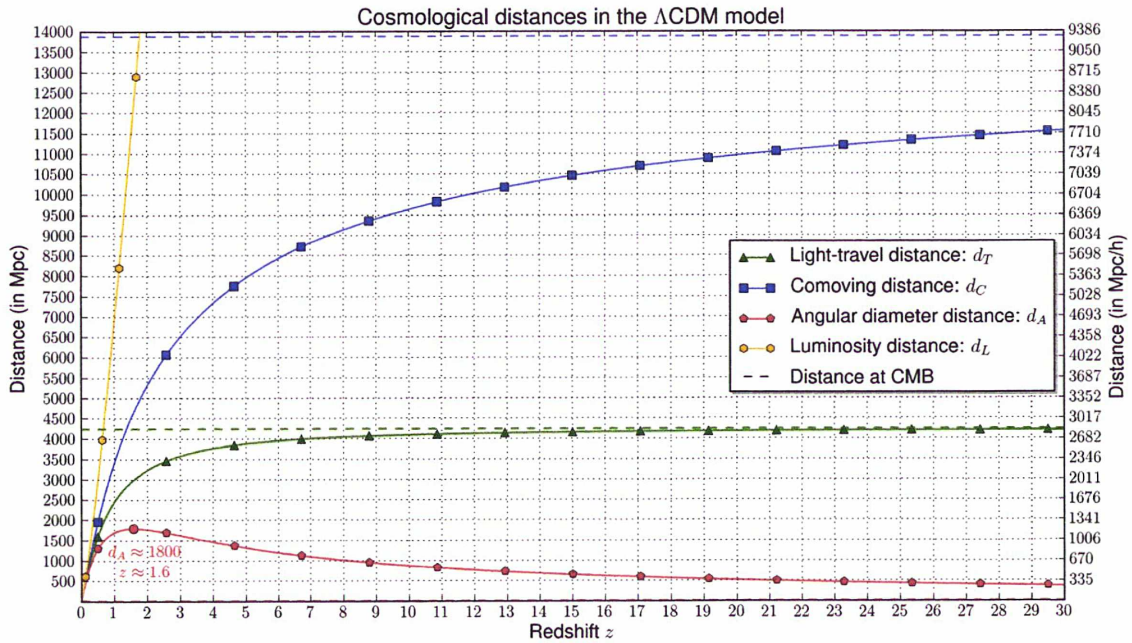


Figure 9 – Évolution des différentes distances cosmologiques pour le modèle de concordance. Connaissant $a(t)$ dans le cas homogène (voir figure 8), il devient possible de calculer les principales distances utilisées en cosmologie : la distance de voyage de la lumière d_T, les distances comobiles parallèle d_C et transverse d_M (équivalentes dans le cas $\Omega_k = 0$), la distance de diamètre angulaire d_A et la distance luminosité d_L. La distance de voyage de la lumière correspond juste au temps de regard vers le passé multiplié par la vitesse de la lumière et l'on a donc : $\lim_{z\to+\infty} d_L(z) \approx 13.82\,\text{al} \approx 4238\,\text{Mpc}$. La distance comobile correspond à la distance qui sépare deux objets en faisant abstraction de l'expansion : elle vaut $d_A(z_{CMB}) \approx 13\,880\,\text{Mpc} \approx 9305\,\text{Mpc/h}$ entre un observateur et le CMB qu'il détecte. La distance de diamètre angulaire, elle, reflète la distance apparente des ob-jets dans le ciel. En ΛCDM, elle est maximale en $z \approx 1.6$. Le fait que $d_A(z_1) = d_A(z_2)$ pour deux sources S_1 et S_2 de même taille physique L signifie que ces deux sources seront vues sous le même angle par l'observateur. C'est par exemple le cas ici pour $z_1 = 1$ et $z_2 \approx 2.5$. Enfin, la distance luminosité d_L, qui s'exprime comme le rapport d'une luminosité intrinsèque sur un flux observé peut se déduire directement de la distance de diamètre angulaire par la relation de dualité d'Etherington 1.27, indépendante de la cosmologie.

question qui sera au cœur des développements abordés dans les chapitres suivants.

1.3 Au-delà de la constante cosmologique

1.3.1 Validité du modèle de concordance et alternatives

Forces et faiblesses de ΛCDM

En explorant les voies ontologiques de la matière noire et de la constante cosmologique, ΛCDM reste jusqu'à présent en très bon accord avec l'ensemble des données observationnelles en utilisant un nombre de paramètres minimal. Et c'est justement cet aspect économique qui peut se résumer basiquement en Ω_{DM} et Ω_Λ qui fait toute sa force. Mais alors où est le problème ? Le problème est toujours le même : celui de la nature de ces deux composantes. Et lorsque que les hypothèses centrales d'un modèle deviennent problématiques, cela ouvre naturellement des questions de nature épistémologique.

Proposer une théorie alternative demande donc, à minima, de parvenir à reproduire l'ensemble des observations indépendantes dont ΛCDM fournit un cadre d'interprétation. A titre d'exemple, des théories comme MoND[22] (Modified Newtonian Dynamics) parviennent à mieux reproduire la dynamique galactique sans recourir à la matière noire, mais rencontrent un grand nombre de difficultés à plus grande échelle. D'autres approches, comme celle des modèles de quintessence s'appuient sur

des modélisations en termes de champs scalaires pour tenter de solutionner certains problèmes liés à la constante cosmologique et notamment le problème de la coïncidence[23]. En effet, pour que les paramètres de densité de la matière et de l'énergie noire et pour que ceux de la matière noire et de la matière baryonique soient aujourd'hui du même ordre de grandeur comme illustré en figure 7, cela demande un réglage très fin de la constante cosmologique du fait d'une très grande dépendance aux conditions initiales. En exhibant des propriétés de « tracking », les modèles de quintessence apportent une solution à ce problème : l'équation d'état de l'énergie noire varie alors au cours de temps, suivant celle du rayonnement avant l'équivalence rayonnement-matière et se détachant de celle-ci par la suite, entraînant le comportement de l'énergie noire observé aujourd'hui. Une telle modélisation en termes de champs scalaires permet également d'asseoir cette énergie noire sur des bases de physique des particules, ce qui n'est pas le cas pour la constante cosmologique.

De l'interprétation des données observationnelles

Une autre alternative consisterait à fournir un cadre d'interprétation différent à certaines preuves observationnelles. Car, au final, nous ne mesurons pas directement la distance aux supernovæ : tout ce que nous mesurons, c'est un redshift et une variation de flux dans le temps. Variation de flux qui est ensuite réinterprété pour en extraire la distance luminosité, puis pour calculer l'évolution du facteur d'échelle. Un problème d'interprétation de cette variation, même si cela est peu probable, pourrait donc entraîner une évaluation erronée de la cosmologie. De même l'interprétation cosmologique du fond diffus est déjà une interprétation. Il ne s'agit pas de remettre en cause ces interprétations, il s'agit juste d'en avoir conscience pour apprécier pleinement les forces et les faiblesses de la cosmologie standard.

Enfin un dernier point concerne non pas les hypothèses de ΛCDM, mais les hypothèses des modèles FLRW en général : le principe cosmologique ou autrement dit l'homogénéité et l'isotropie. Au niveau local, le principe cosmologique est bien entendue erroné mais à partir d'une certaine échelle, la distribution de matière semble effectivement relativement homogène et isotrope. Considérer que l'Univers est en moyenne friedmannien à grande échelle semble donc tout à fait fondé pour en déduire sa dynamique. Sauf qu'en toute rigueur, il n'y a pas commutativité de cette opération de moyennage[24] en relativité générale : moyenner la distribution de matière pour y effectuer des calculs de nature cosmologique n'est pas équivalent à effectuer ces mêmes calculs au sein de l'Univers tel qu'il est puis de moyenner le résultat. Cela peut entraîner naturellement deux types d'effets : une mauvaise interprétation des observations d'une part et une mauvaise évaluation de la dynamique de l'Univers d'autre part. Le premier point trouve son origine dans le fait que lorsqu'une distance luminosité est mesurée, celle-ci est souvent interprétée directement dans un cadre rigoureusement homogène pour en déduire la cosmologie. Or la géodésique nulle entre la source et l'observateur n'est pas celle d'une métrique FLRW, ou dit autrement, la lumière n'a pas voyagé en ligne droite du fait de la structuration. C'est justement à ce problème que seront consacrés les chapitres suivants. Le second point, lui, est lié à toute la thématique de la backreaction qui s'intéresse aux rétroactions de la structuration de la matière sur la dynamique de l'Univers elle-même. Car, là aussi, la dynamique d'un univers friedmannien et la dynamique d'un univers en moyenne friedmannien ne sont pas équivalentes. Et dans un contexte où l'énergie noire devient d'autant plus nécessaire pour reproduire les observations que l'Univers se structure, la question de l'effet des inhomogénéités locales sur la dynamique globale s'avère être plus que pertinente. Mais si l'on sait que ces effets existent, la question de leur amplitude reste aujourd'hui ouverte.

1.3.2 Quintessence et autres équations d'état

Équations d'état différentes de -1 : WCDM et WACDM

La réinterprétation de la constante cosmologique comme un fluide d'énergie noire et non comme une constante de nature purement géométrique peut conduire à considérer des équations d'état w_{DE} constantes mais différentes de $w_\Lambda = -1$. Ces modèles sont regroupés sous le sigle WCDM. Pour produire un comportement accéléré, il suffit que la condition suivante soit vérifiée :

$$\boxed{w_{DE} < -\frac{1}{3}} \tag{1.31}$$

Lorsque w_{DE} est strictement inférieure à -1, on parle alors de « phantom dark energy »[25] : dans ce cas la densité d'énergie de l'Univers augmente avec le temps et l'accélération de l'expansion ne cesse de croître, conduisant à un « Big Rip » en un temps fini[26]. Ces modèles ne sont pour l'instant pas exclus par les observations[7] et l'on considèrera le cas $w_{DE} = -1.2$ dans la suite.

Pour aller plus loin, il est également possible de relaxer le caractère constant de l'équation d'état. On parle alors de modèles WACDM pour signifier la dépendance de w_{DE} en termes de facteur d'échelle. L'une des modélisations les plus simples et les plus souvent considérées dans le cadre de l'analyse du fond diffus cosmologique est :

$$\boxed{w_{DE}(a) = w_0 + w_a(1-a)} \tag{1.32}$$

Mais ce type de modélisation reste effective. Une façon de l'asseoir sur des bases théoriques plus solides est de considérer des modèles de quintessence.

Modèles de quintessence : RPCDM et SUCDM

Comme précisé précédemment, la constante cosmologique entraîne mécaniquement le problème de la coïncidence, de la grande dépendance aux conditions initiales et du caractère *ad hoc* de Λ vu sous l'angle de la physique des particules[27]. Les modèles de quintessence se proposent de conserver l'approche en termes d'énergie noire au problème de l'accélération de l'expansion tout en essayant d'apporter une réponse à ces faiblesses conceptuelles.

Pour cela, la constante cosmologique est remplacée par l'introduction d'un champ scalaire Q de potentiel d'auto-interaction $V(Q)$ dans le tenseur énergie-impulsion qui s'écrit[28], en unités naturelles $c = \bar{h} = 1$:

$$T_{\mu\nu}^{(Q)} = \frac{\partial Q}{\partial \mu}\frac{\partial Q}{\partial \nu} - g_{\mu\nu}\left[\frac{1}{2}g^{\alpha\beta}\frac{\partial Q}{\partial \alpha}\frac{\partial Q}{\partial \beta} + V(Q)\right] \tag{1.33}$$

où Q et $V(Q)$ satisfont l'équation de Klein-Gordon :

$$\ddot{Q} + 3H\dot{Q} + \frac{\mathrm{d}V(Q)}{\mathrm{d}Q} = 0 \tag{1.34}$$

La densité et la pression de la quintessence s'écrivent par conséquent :

$$\rho_Q = T_{00} = \frac{\dot{Q}^2}{2} + V(Q) \qquad \text{et} \qquad P_Q = T_{ii} = \frac{\dot{Q}^2}{2} - V(Q) \tag{1.35}$$

si bien que son équation d'état, variable au cours du temps, est :

$$\boxed{w_Q = \frac{P_Q}{\rho_Q} = \frac{\frac{\dot{Q}^2}{2} - V(Q)}{\frac{\dot{Q}^2}{2} + V(Q)}} \tag{1.36}$$

Lorsque le terme potentiel $V(Q)$ domine largement le terme cinétique $\frac{\dot{Q}^2}{2}$, on retrouve une équation d'état à l'origine d'un comportement accéléré :

$$\frac{\dot{Q}^2}{2} \ll V(Q) \qquad \Rightarrow \qquad w_Q \simeq -1 < -\frac{1}{3} \tag{1.37}$$

Spécifier un modèle de quintessence revient donc à spécifier la forme du potentiel qui pilotera l'évolution de l'équation d'état au cours de l'évolution de l'Univers. Et pour cela, plusieurs solutions standard existent.

En effet, deux potentiels abondamment étudiés dans la littérature sont les potentiels de Ratra-Peebles[29] (RP) et de supergravité[30] (SUGRA ou SU), le second apportant une légère correction au premier. En plus de supprimer la dépendance aux conditions initiales et donc d'éviter le problème de coïncidence, ces formes de potentiels sont compatibles avec la physique des hautes énergies[31, 32]. Ils s'écrivent :

$$\boxed{V_{RP}(Q) = \frac{\lambda^{\alpha+4}}{Q^\alpha}} \qquad \text{et} \qquad \boxed{V_{SU}(Q) = \frac{\lambda^{\alpha+4}}{Q^\alpha} e^{4\pi G Q}} \tag{1.38}$$

où $\alpha \geq 0$ et λ sont des paramètres libres.

Résumé des modèles considérés

Il ne s'agira pas, dans la suite, de débattre de la pertinence et du bien-fondé de ces différentes modélisations car ce n'est pas le sujet. Le sujet est plutôt d'explorer l'empreinte de différentes cosmologies « réalistes », c'est-à-dire compatibles avec les observations, sur la formation des grandes structures et la détermination des distances cosmologiques. C'est dans ce cadre, que l'on considèrera WCDM, RPCDM et SUCDM comme autant de façons de s'écarter du modèle de concordance ΛCDM et ainsi d'étudier des dynamiques légèrement différentes. L'objectif n'est donc en aucun cas de proclamer la validité d'un modèle par rapport à tel autre, mais plutôt de qualifier et de quantifier les conséquences observationnelles d'énergies noires de nature différente.

La figure 10 résume ainsi les paramètres utilisés dans le cadre du projet Dark Energy Universe Simulation abrévié DEUS dans la suite. Ces paramètres permettent de sonder le comportement d'énergies noires dont l'équation d'état est différente de -1. Dans un cadre purement homogène, cela entraîne déjà des différences de l'ordre de quelques pourcents sur l'évolution du facteur d'échelle et des distances cosmologiques comme illustré en figure 11. Dans un cadre non-linéaire, les effets pourraient être beaucoup plus subtils, l'énergie noire laissant son empreinte à tous les niveaux de tels processus non-linéaires. Aussi, pour comprendre ces effets, et pour mieux comprendre ce que nous disent réellement les observations sur le problème de l'accélération de l'expansion, l'outil numérique va s'avérer être d'une grande aide pour sonder ces processus. Et c'est justement ce dont il s'agira dans les prochains chapitres.

1.3.3 Développements autour de la gravité de Brans-Dicke

Mais avant de commencer à sonder les processus non-linéaires d'effondrement de la matière, faisons un petit détour du côté de la théorie de Brans-Dicke. En effet, si la quintessence permet déjà de dépasser la constante cosmologique par l'introduction de champs scalaires, il est possible d'aller plus loin avec les théories de type scalaire-tenseur. Cette classe de théories peut être vue comme la généralisation la plus simple de la relativité générale : l'interaction gravitationnelle y est transmise par la métrique, comme en relativité générale, mais aussi par une composante scalaire. Historiquement, ce type de théories scalaire-tenseur a été introduit à la fois pour pouvoir explorer des généralisations à plus de 4 dimensions d'espace-temps comme chez Jordan, et à la fois pour réconcilier la relativité générale avec le principe de Mach en permettant une variation des constantes fondamentales, comme chez

Cosmological parameters of the Dark Energy Universe Simulation project								
	h_0	Ω_{b_0}	Ω_{M_0}	Ω_{DE_0}	w_0	w_a	n_s	σ_8
LCDMW5	0.72	4.385×10^{-2}	0.26	0.74	-1	0	0.963	0.79
RPCDMW5	0.72	4.385×10^{-2}	0.23	0.77	-0.87	0.08	0.963	0.66
SUCDMW5	0.72	4.385×10^{-2}	0.25	0.75	-0.94	0.19	0.963	0.73
LCDMW7	0.72	4.356×10^{-2}	0.2573	0.7426	-1	0	0.963	0.801
RPCDMW7	0.72	4.385×10^{-2}	0.23	0.77	-0.87	0.08	0.963	0.66
WCDMW7	0.72	4.356×10^{-2}	0.275	0.725	-1.2	0	0.963	0.852

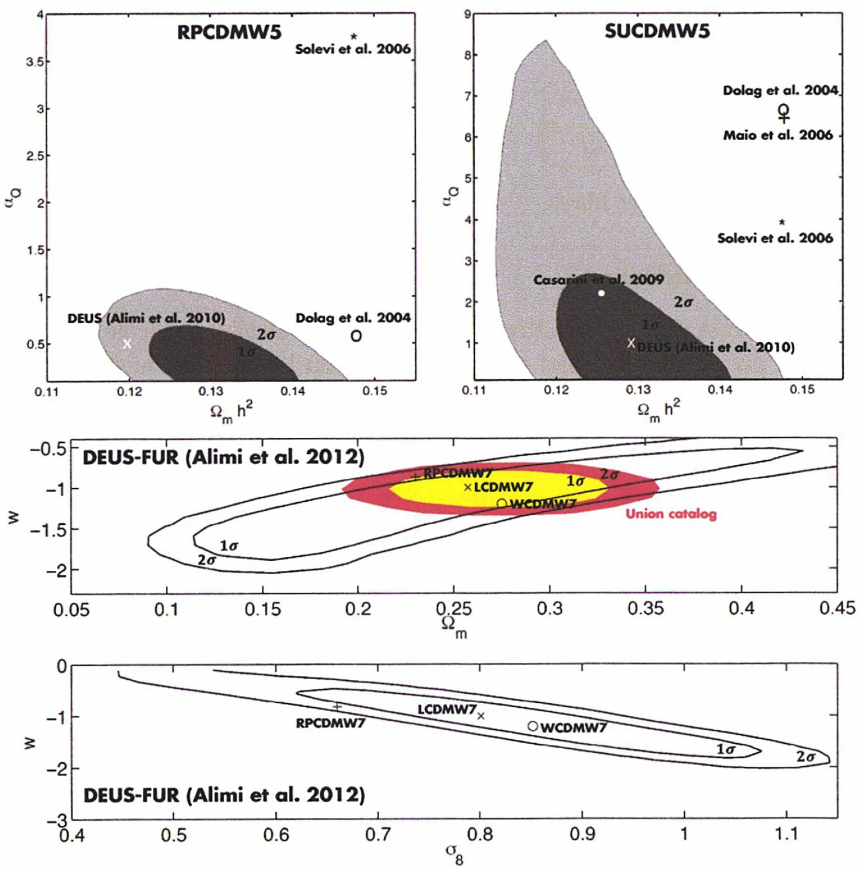


Figure 10 – Paramètres cosmologiques du projet Dark Energy Universe Simulation (DEUS). La table résume l'ensemble des paramètres choisis pour le projet Dark Energy Universe Simulation : LCDM pour le modèle de concordance avec constante cosmologique, RPCDM pour un modèle de quintessence Ratra-Peebles, SUCDM pour un modèle de quintessence de supergravité et WCDM pour un modèle « phantom » avec une équation d'état inférieure à -1. Ces modèles viennent en deux versions suffixées W5 ou W7 suivant qu'ils sont basés sur les données de WMAP5 ou de WMAP7. Dans cette table, h_0 représente le paramètre de Hubble adimensionné 1.6, Ω_{b_0}, $\Omega_{M_0} = \Omega_{b_0} + \Omega_{DM_0}$ et Ω_{DE_0} représentent les valeurs des paramètres de densité des baryons, de la matière et de l'énergie noire sachant que tous les modèles sont choisis ici de courbure nulle $\Omega_k = 0$, w_0 et w_a modélisent l'équation d'état de l'énergie noire suivant l'équation 1.32, $n_s = \frac{\mathrm{d} \ln P(k)}{\mathrm{d} \ln k} + 1$ est l'indice spectral intervenant dans la modélisation du spectre des fluctuations primordiales en loi de puissance[33] et où σ_8 est la moyenne quadratique des fluctuations de la distribution de densité dans des sphères comobiles de 8 Mpc/h de rayon[34]. Ces modèles sont considérés comme « réalistes » dans le sens où leurs paramètres ont été choisis pour être compatibles avec les données observationnelles. En effet, les quatre figures illustrent le choix de certains de ces paramètres dans le plan $\Omega_{M_0} h_0^2$ / α pour les modèles de quintessence[35] et dans les plans $\Omega_{M_0} h_0^2$ / w_0 et σ_8 / w_0 pour les modèles basés sur WMAP7[36]. La plupart des paramètres sont dans les ellipses de confiance à 1σ fournies par les contraintes observationnelles. Aussi, ces différents modèles vont permettre de tester l'effet de la cosmologie sur la formation des grandes structures au travers de simulations numériques.

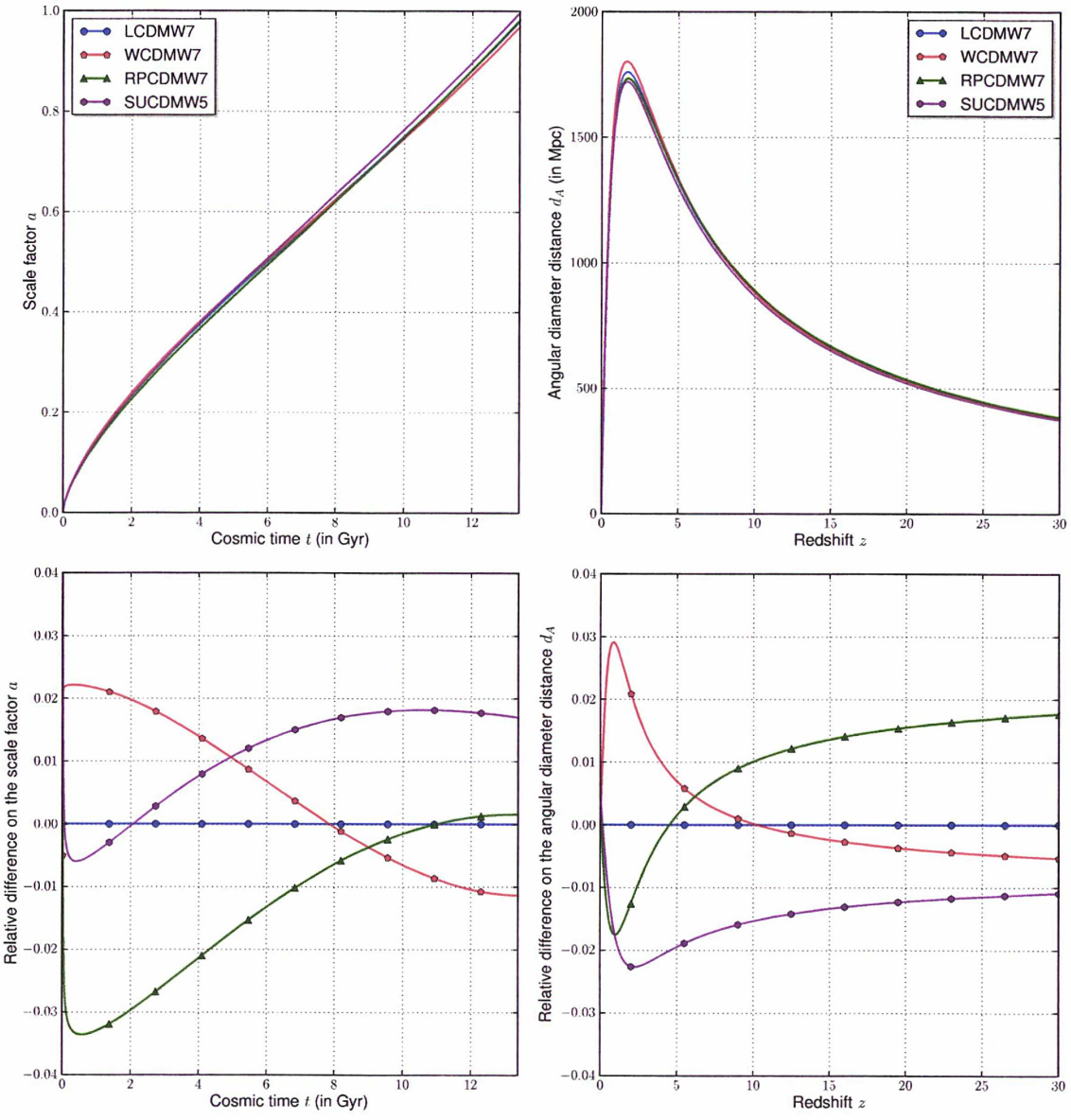


FIGURE 11 – Dynamique des modèles cosmologiques du projet Dark Energy Universe Simulation (DEUS). La dynamique de quatre des modèles dont les paramètres sont listés en figure 10 est présentée : à gauche il s'agit de l'évolution du facteur d'échelle en fonction du temps cosmique, tandis qu'à droite, il s'agit de la distance de diamètre angulaire en fonction du redshift. Étant donné que les modèles sont tous en accord avec les relevés observationnels, les différences sont minces. Aussi, les graphiques du bas présentent les différences relatives des modèles par rapport au cas de référence LCDMW7 : à gauche il s'agit de $\frac{a-a^{(\mathrm{LCDMW7})}}{a^{(\mathrm{LCDMW7})}}$ et à droite de $\frac{d_A-d_A^{(\mathrm{LCDMW7})}}{d_A^{(\mathrm{LCDMW7})}}$. Ainsi, dans un cas comme dans l'autre, les différences sont de l'ordre de quelques pourcents. Toutefois, cette dynamique légèrement différente va laisser son empreinte dans la formation des structures, qui va à son tour influer sur la mesure des distances. L'étude de ces modèles réalistes au sens observationnel va donc permettre de sonder les effets d'une énergie noire de nature différente de la constante cosmologique.

Brans[37] et Dicke[38]. Ce faisant, la théorie de Brans-Dicke respecte le principe d'équivalence faible, mais pas le principe d'équivalence fort. Cette approche est particulièrement intéressante pour deux raisons : d'une part, elle permet de réinterpréter les problématiques liées à l'énergie noire comme un effet de référentiel[39, 40, 41] et d'autre part, elle permet de traiter des scénarii inflationnaires dans l'Univers primordial.

Un travail original, notamment de nature numérique, portant sur l'hypothèse AWE[42], une généralisation des théories scalaire-tenseurs envisageant des couplages non universels a été effectué dans le cadre de cette thèse et est présenté en annexe. En partant d'une action dans « Jordan Frame » [8] sous la forme :

$$\begin{aligned} S_{JF} =& \frac{c^3}{16\pi\tilde{G}} \int \mathrm{d}^4x\sqrt{-\tilde{g}} \left(\Phi\tilde{R} - \frac{\omega_{BD}(M(\Phi))}{\Phi} \tilde{g}^{\mu\nu} \frac{\partial\Phi}{\partial x^\mu} \frac{\partial\Phi}{\partial x^\nu} \right) \\ &+ \varepsilon_1 S_m \left[\psi_m, \tilde{g}_{\mu\nu}\right] + \varepsilon_2 S_a \left[\psi_a, M(\Phi)^2 \tilde{g}_{\mu\nu}\right] \end{aligned} \tag{1.39}$$

où :

- $\tilde{G}$ est la constante de gravitation « nue », qui diffère de la constante de gravitation effective G_{eff}
- $\tilde{g}_{\mu\nu}$ est la métrique du référentiel de Jordan couplée au champ de matière ordinaire
- $\tilde{g}$ est son déterminant
- $\tilde{R}$ est le scalaire de courbure qui lui est associé
- Φ est le degré de liberté scalaire tel que la constante de gravitation observée vaut $G_{\text{eff}} = \frac{\tilde{G}}{\Phi}$
- ω_{BD} est la fonction de couplage de Brans-Dicke
- S_m est l'action de la matière, reposant sur le champ ψ_m
- S_a est celle du secteur AWE, reposant sur le champ ψ_a
- ε_1 et ε_2 représentent les signes des termes cinétiques des champs scalaires : $\varepsilon = +1$ modélise un champ classique, $\varepsilon = -1$ un champ « phantom »

avec :

$$\tilde{g}_{\mu\nu} = A_m^2(\varphi) g_{\mu\nu} \qquad \text{et} \qquad \begin{cases} \Phi = \frac{\tilde{G}}{G} = A_m^{-2}(\varphi) \\ (3 + 2\omega_{BD})^2 = \frac{\mathrm{d}\ln A_m(\varphi)}{\mathrm{d}\varphi} \\ M(\Phi) = \frac{A_a(\varphi)}{A_m(\varphi)} \end{cases} \tag{1.40}$$

où $A_m(\varphi) = e^{k_m\varphi}$ et $A_a(\varphi) = e^{-k_m\varphi}$ sont les fonctions de couplage non-minimal pour le secteur visible et invisible et en s'intéressant au cas cosmologique d'une métrique FLRW de courbure nulle pour laquelle :

$$E_\varphi = \frac{1}{2}\left(\left(\frac{\mathrm{d}\varphi}{\mathrm{d}\tau} \right)^2 + V(\varphi) \right) \qquad \text{et} \qquad V(\varphi) = \varepsilon_1 \frac{C_m^2}{A_m^2(\varphi)} + \varepsilon_2 \frac{C_a^2}{A_a^2(\varphi)} \tag{1.41}$$

où C_m et C_a sont des constantes du mouvement, il est possible d'obtenir une grande variété de comportements d'expansion accélérée comme illustré en figure 12. L'exploration de ces comportements est décrite plus en détail dans l'annexe et l'article[43] qui l'accompagne.

Comme nous l'avons vu, il existe de multiples manières de modéliser une accélération de l'expansion au-delà de la constante cosmologique sur diverses bases théoriques comme la quintessence ou la théorie de Brans-Dicke. Toutefois, l'objectif dans la suite n'est pas de se concentrer sur l'Univers homogène ou sur les fondements théoriques de tous les modèles au-delà de ΛCDM. La théorie est prolifique sur le sujet et la littérature abondante. Non, l'objectif est plutôt de comprendre comment des histoires d'expansion légèrement différentes de celle du modèle de concordance et compatibles avec les

8. Voir détails en annexe.

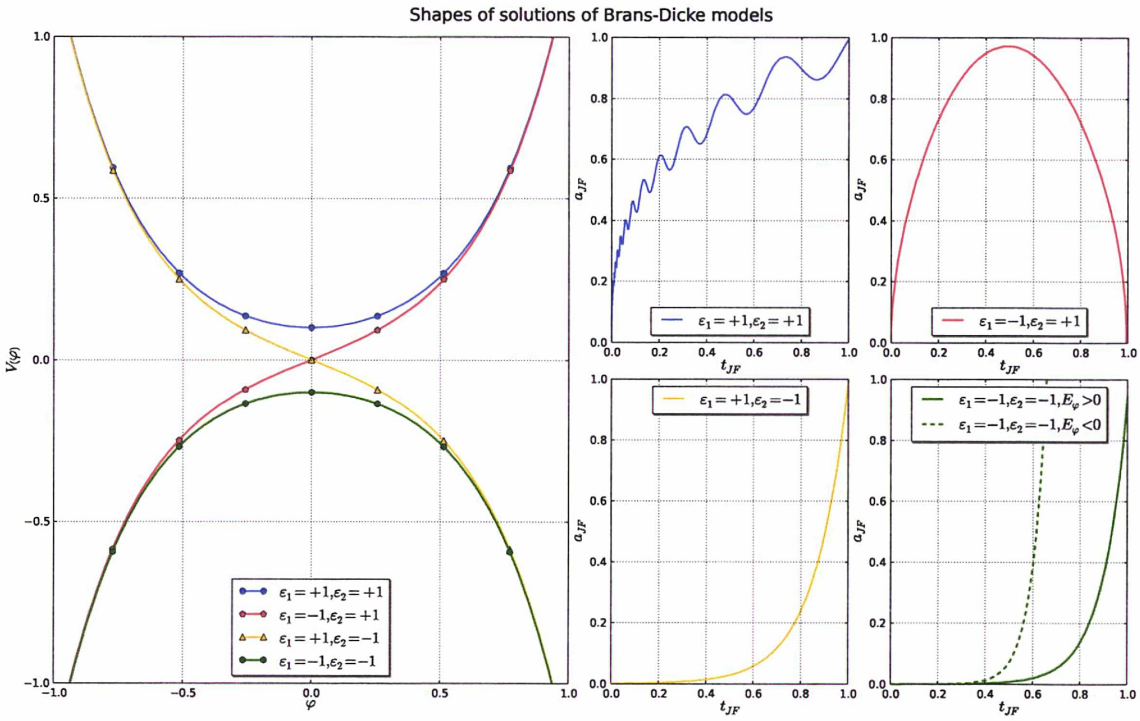


FIGURE 12 – Profil des solutions numériques du modèle de Brans-Dicke considéré. À gauche, les symétries de $V(\varphi)$ sont déterminées par les valeurs de ε_1 et ε_2 tandis que son profil est piloté par les valeurs de C_m, C_a, A_m et A_a. Il est alors possible d'explorer pour cinq types de solutions comme présenté à droite, où a_{EF} est le facteur d'échelle dans le référentiel d'Einstein. Le cas des ε positifs, correspondant à des termes cinétiques ordinaires pour la matière et le secteur AWE, permet d'obtenir des solutions d'expansion oscillante. Lorsque la matière est associée à un champ scalaire « phantom », l'Univers se recontracte rapidement après une phase initiale d'expansion. Les trois autres cas, c'est-à-dire lorsqu'un champ scalaire « phantom » est associé au secteur AWE, conduisent à l'obtention de profils très accélérées.

observations actuelles pourraient influer sur la formation des grandes structures et sur l'interprétation des observations elles-mêmes. Car si l'accélération de l'expansion laisse sa trace dans tous ces phénomènes, alors les mesures effectuées sur ces mêmes phénomènes devraient permettre d'en comprendre davantage sur la nature et l'origine de cette accélération.

Chapitre 2

L'Univers structuré : effondrement gravitationnel et sondes cosmologiques

***

2.1 Formation des grandes structures 26
2.1.1 Grands relevés observationnels et Univers local 26
2.1.2 Fluctuations primordiales de densité et régime linéaire 28
2.1.3 Régime non-linéaire et scénario hiérarchique de formation 30
2.2 Simuler l'évolution des grandes structures 31
2.2.1 Fondements et principes des simulations cosmologiques 31
2.2.2 Caractérisation des champs et des halos de matière noire 33
2.2.3 Note sur le rôle des baryons 36
2.3 Sonder la cosmologie au moyen de la structuration 38
2.3.1 Empreinte de la cosmologie sur la formation des structures avec une statistique maximale 38
2.3.2 Propriétés du champ de matière : l'exemple du spectre de puissance 38
2.3.3 Les oscillations acoustiques des baryons : un échelle des distances standard ? . 39
2.3.4 Propriétés des halos, l'exemple de la fonction de masse 41
2.3.5 De la nécessité de nouvelles sondes dans l'espace des redshifts 42

En se basant sur la relativité générale, la cosmologie, et plus spécifiquement les modèles de Friedmann-Lemaître, nous conduisent à des modèles d'univers homogènes et isotropes en expansion. Mais la réalité n'est pas homogène et isotrope : les galaxies, les amas et les superamas sont autant d'écarts locaux au principe cosmologique. Ainsi, même si en moyenne, à grande échelle, le principe cosmologique semble respecté, il n'en est pas de même à l'échelle locale : à cette échelle l'Univers apparait comme structuré. Et c'est justement par cette structuration que nous sommes à même de sonder l'Univers. Mais alors, quelle empreinte l'accélération de l'expansion laisse-t-elle dans la formation des structures cosmiques ? Et inversement, comment sonder la nature de l'accélération de l'expansion à partir de l'observation de la structuration à grande échelle ? Au chapitre précédent, nous avons vu que des modèles de quintessence pouvaient donner lieu à des histoires d'expansion légèrement différentes de celle du modèle de concordance tout en restant compatibles avec les contraintes observationnelles actuelles. Aussi, avant de s'intéresser aux conséquences de ces différentes histoires d'expansion sur la propagation de la lumière et donc sur la mesure des distances dans l'Univers il est nécessaire de comprendre la nature des traces laissées par l'expansion dans la formation des structures. C'est l'objet de ce chapitre qui se divise en trois grandes sections. La première présentera le scénario hiérarchique de formation des

grandes structures et les principes qui sous-tendent l'effondrement de la matière. La seconde traitera des simulations cosmologiques et du rôle central qu'elles jouent dans la compréhension de l'empreinte de la cosmologie sur la formation et l'évolution des grandes structures. Il s'agira ici de discuter essentiellement de certains aspects physiques, les aspects numériques et les aspects d'implémentation étant abordés dans la seconde partie de ce manuscrit. La troisième, elle, se focalisera sur les sondes cosmologiques issues de la structuration et notamment sur le spectre de puissance, la fonction de masse et les oscillations acoustiques des baryons. Cette troisième partie s'appuiera notamment sur des résultats issus du projet Dark Energy Universe Simulation. Elle nous conduira enfin à nous interroger sur la nécessité de disposer de nouvelles sondes dans l'espace des redshifts.

2.1 Formation des grandes structures

2.1.1 Grands relevés observationnels et Univers local

La cosmologie observationnelle constitue un point de départ de choix pour aborder les problématiques de structuration. Depuis le grand débat des années 1920 sur la nature des nébuleuses spirales, notre connaissance de l'Univers local a bien progressée. De plus, depuis une vingtaine d'années, les grands relevés systématiques de redshifts nous ont permis de sonder la structuration jusqu'à de très grandes distances. Même si la mise en pratique pose de vrais défis liés à l'instrumentation et à l'analyse des données, le principe central de ces catalogues peut être résumé assez simplement : mesurer le redshift et la position angulaire de tous les objets astrophysiques dans une portion de ciel jusqu'à une magnitude donnée. En procédant de la sorte, il est ainsi possible de reconstruire la distribution de la matière lumineuse à grande échelle, de part et d'autre de la zone d'évitement (« Zone Of Avoidance ») constitué par le disque galactique. Et cette distribution est loin d'être homogène localement, même si au-dessus d'échelles typiques de l'ordre de 10^2 Mpc/h, l'Univers redevient statistiquement homogène. En effet, au-delà de la Voie Lactée, du Groupe Local et du superamas de la Vierge dont l'environnement est présenté en figure 14, les catalogues de redshifts laissent apparaître la structure filamentaire de la distribution des galaxies illustrée en figure 13. Des projets observationnels comme le Sloan Digital Sky Survey[44] ont permis de dresser une cartographie de l'Univers à grande échelle avec 1.5 millions de galaxies jusqu'à $z \approx 0.7$ et plus d'une centaine de milliers de quasars entre $z \approx 2.2$ et $z \approx 3.5$. À l'horizon 2020, la mission spatiale Euclid[45] commencera son relevé méthodique basé sur l'analyse du lentillage gravitationnel pour tracer la distribution de matière sur environ 15 000 degrés carrés et jusqu'à $z \approx 2$.

Les structures qui apparaissent à l'issue de ces mesures peuvent se décomposer en éléments de différentes dimensionnalité : des « cosmic sheet » 2D, des filaments 1D d'une longueur typique de 50 Mpc/h à 80 Mpc/h et des superamas tridimensionnels. Ces surdensités sont séparées par des sous-densités : les vides cosmiques. Des structures largement plus grandes que l'échelle supposée d'homogénéité, et qui viennent donc questionner la validité du principe cosmologique, ont également été détectées, comme dernièrement le Grand Mur de Hercule - Couronne boréale[46] qui semble s'étendre sur plus de 2 Gpc. La « cosmic web » constituée de tous ces éléments n'est pas inerte par rapport à l'expansion : depuis leur formation, ces grandes structures ont évolué et continuent d'évoluer. Ainsi, les galaxies, les amas, et les superamas sont doués de mouvements propres, de vitesses péculières. Localement, l'analyse des champs de vitesse semble indiquer que la Galaxie, le Groupe Local et le superamas de la Vierge sont attirés par le Grand Attracteur, situé dans le superamas de l'Hydre-Centaure. Ces champs de vitesse conduisent à former, sur le temps long, des structures de plus en plus grandes et de plus en plus massives, isolées par des vides de plus en plus grands eux aussi.

Cette structure filamentaire, qui apparait bien sur l'élément de gauche de la figure 13 a deux conséquences directes sur la métrique et donc sur la cosmologie. La première est de nature cinématique et touche à l'Univers Observable. En effet, cette succession de sous-densités et de surdensités autour de

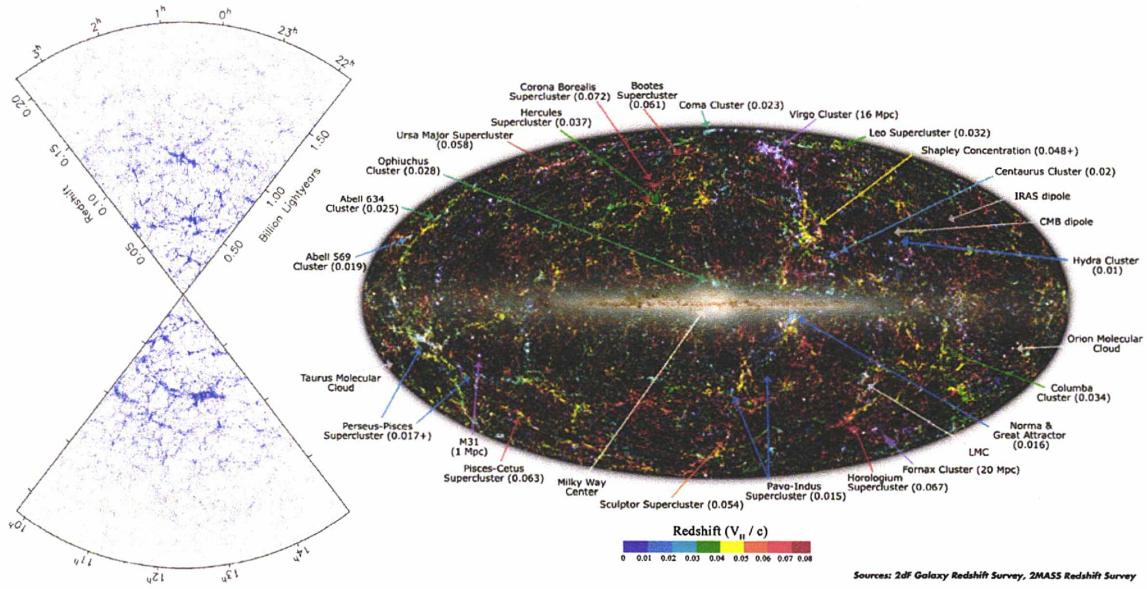


FIGURE 13 – Structuration à grande échelle déduite des catalogues de redshifts 2dF et 2MRS. Chaque point correspond à la mesure d'un redshift. A gauche, est représentée la distribution de galaxies jusqu'à $z \approx 0.2$ obtenue à partir du 2dF Galaxy Redshift Survey[47], qui a mesuré le redshift de près de 250 000 objets dans une zone totale de 1500 degrés carrés de part et d'autre du disque galactique entre 1997 et 2002. À droite les grandes structures de l'Univers local sont indiquées au sein des données du 2MASS Redshift Survey[48]. Dans les deux cas, la structure filamentaire apparait très clairement. Dans les données 2MRS, les amas et superamas locaux ressortent de la structure, comme par exemple l'amas de la Vierge, situé en haut de l'image, et qui constitue le noyau du superamas de la Vierge dont fait partie le Groupe Local. Cet amas, situé à environ 16 Mpc contient de 1300 à 2000 galaxies pour une masse totale estimée à $1.2 \times 10^{15}\,\mathrm{M}_\odot$[49].

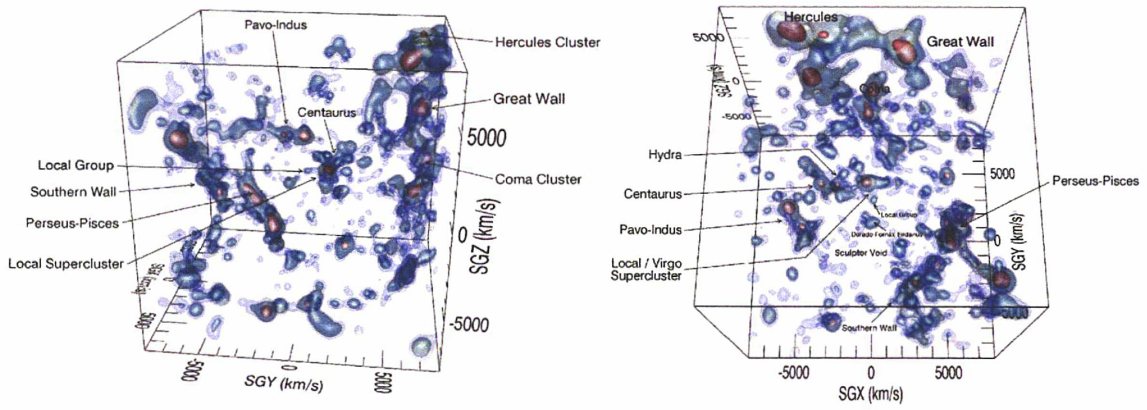


FIGURE 14 – Reconstruction de la distribution locale de la matière par Courtois *et al.*, 2013[50]. Ces deux vues présentent une reconstruction de la densité de matière locale et permettent de localiser la situation du Groupe Local au sein de la structure illustrée en figure 13. Les axes SGX, SGY, SGZ constituent une version cartésienne du système de coordonnées supergalactique[51]. L'unité utilisée correspond à l'espace des redshifts en supposant une constante de Hubble $H_0 = 74\,\mathrm{km/s/Mpc}$. La densité est estimée à partir d'une compilation de catalogues d'environ 30 000 galaxies. Les 3 couleurs correspondent à des seuils de 0.3, 0.1 et 0.07 $L^\star$ galaxies par $(100\,\mathrm{km/s})^3$, où $L^\star$ est la luminosité caractéristique associée à la magnitude absolue $M^0_{B(0)} = -20.6$[52]. Le superamas de la Vierge apparait au centre de la boîte. Celui-ci est constitué d'environ 10^4 galaxies et s'étend dans une sphère d'environ 33 Mpc de diamètre. Sur le côté, le Great Wall et l'amas d'Hercules, qui est situé à $z \approx 0.037$, font partie des plus grandes structures de l'Univers local. La structure filamentaire délimite également de larges domaines sous-denses qui constituent les vides cosmiques, comme celui délimité par le Southern Wall.

Position of the firsts CMB peaks and troughs according to WMAP		
	l	ΔT^2 [μK^2]
First peak	220.8 ± 0.7	5624 ± 30
First trough	412.4 ± 1.9	1726 ± 28
Second peak	530.9 ± 3.8	2485 ± 44
Second trough	675.2 ± 11.1	1688 ± 81

Figure 15 – Position et amplitude des premiers pics et creux du spectre de puissance des fluctuations de température d'après WMAP[55]. Ces valeurs peuvent être reliées aux paramètres cosmologiques et sont donc utilisées pour contraindre les modèles.

l'observateur que nous sommes influe sur la propagation de la lumière et donc sur notre perception de l'Univers. Cela induit des effets comme le lentillage gravitationnel, l'effet Sachs-Wolfe intégré et vient modifier localement la relation qui lie le redshift au facteur d'échelle. Tous ces effets seront discutés dans les chapitres suivants. Mais cela pourrait également avoir des effets dynamiques, et c'est tout l'objet de propositions comme la conjecture de la backreaction[53, 24, 54] : en effet, la structuration influe sur la métrique et donc sur le facteur d'expansion local, conduisant les vides cosmiques à subir des taux d'expansion plus élevés que ceux des filaments. Toutefois l'amplitude de ces effets reste aujourd'hui une question ouverte qui dépasse largement du cadre de cette thèse.

2.1.2 Fluctuations primordiales de densité et régime linéaire

Spectre de puissance du fond diffus cosmologique

L'origine de la structure filamentaire observée aujourd'hui peut être retracée jusqu'à l'Univers primordial, notamment par la mesure des anisotropies du fond diffus cosmologique. À cette époque, qui correspond à un redshift de $z_{\mathrm{CMB}} \approx 1090.48$ selon les dernières analyses de Planck[7], les écarts de température sont de l'ordre de :

$$\frac{\delta T}{T} \approx 10^{-5} \tag{2.1}$$

Ces fluctuations primordiales de température, et donc de densité, constituent les germes de la formation des grandes structures. Leur analyse passe par la caractérisation du spectre de puissance angulaire des fluctuations de température $\Theta(\vec{n})$ avec $\vec{n}$ le vecteur associé à la position (θ, φ) sur le ciel :

$$\Theta(\vec{n}) = \frac{T(\vec{n}) - T_0}{T_0} = \sum_{l=0}^{+\infty} \sum_{m=-l}^{+l} a_{lm} Y_{lm}(\vec{n}) \qquad \text{avec} \qquad T_0 = \langle T \rangle \approx 2.7255\,\mathrm{K} \tag{2.2}$$

et où les $a_{lm}Y_{lm}$ représentent la décomposition de $\Theta(\vec{n})$ en harmoniques sphériques. En supposant la gaussianité des fluctuations, on a $\langle a_{lm} \rangle = 0$. On peut, de plus, définir le spectre de puissance angulaire par :

$$\boxed{C_l = \left\langle |a_{lm}|^2 \right\rangle = \frac{1}{2l+1} \sum_m \left\langle |a_{lm}|^2 \right\rangle} \qquad \text{avec} \qquad \left\langle \Theta^2 \right\rangle = \sum_l \frac{2l+1}{4\pi} C_l \tag{2.3}$$

Ce spectre est au cœur des analyses de missions comme WMAP et Planck la position des pics et des creux permet de contraindre efficacement la valeur des paramètres cosmologiques, notamment via la distance angulaire à la surface de dernière diffusion. La position et l'amplitude des premiers pics de ce dernier sont donnés en figure 15.

Évolution linéaire des surdensités

Dans un premier temps, les fluctuations de densité associées aux fluctuations de température vont évoluer linéairement. Dans la suite, on caractérise les sous-densités et les surdensités par le contraste

de densité :

$$\boxed{\delta\left(\vec{x},t\right)=\frac{\rho\left(\vec{x},t\right)-\bar{\rho}\left(t\right)}{\bar{\rho}\left(t\right)}} \tag{2.4}$$

où ρ est la densité locale et où $\bar{\rho}$ est la densité moyenne de l'Univers. Dans l'ère de matière, l'évolution de δ peut alors être décrite par les équations de la dynamique newtonienne pour un fluide parfait perturbé dans un Univers friedmannien en expansion en coordonnées comobiles[56] :

$$\boxed{\begin{aligned}&\frac{\partial\delta}{\partial t}+\frac{1}{a}\vec{\nabla}\cdot\left[\left(1+\delta\right)\vec{v}\right]=0\\&\frac{\partial\vec{v}}{\partial t}+\frac{1}{a}\left(\vec{v}\cdot\vec{\nabla}\right)\vec{v}+\frac{\dot{a}}{a}\vec{v}=-\frac{1}{a}\vec{\nabla}\Phi-\frac{1}{a\rho}\vec{\nabla}P\\&\nabla^2\Phi=4\pi G\rho a^2\delta\end{aligned}} \tag{2.5}$$

avec, dans l'ordre, la conservation de la masse, l'équation d'Euler et l'équation de Poisson et où $\vec{v}$ est la vitesse locale du fluide, P sa pression et Φ le potentiel gravitationnel. Dans le régime linéaire, il est possible de décomposer δ en des modes croissants D_+ et décroissants D_- :

$$\boxed{\begin{aligned}&\delta\left(\vec{x},t\right)=D_+\left(t\right)\varepsilon_+\left(\vec{x}\right)+D_-\left(t\right)\varepsilon_-\left(\vec{x}\right)\\&D_+\left(a\right)=\frac{5}{2}\Omega_{M_0}\frac{H\left(a\right)}{H_0}\int_0^a\left(\frac{H_0}{a'H\left(a'\right)}\right)^3\mathrm{d}a'\\&D_-\left(a\right)=H\left(a\right)\end{aligned}} \tag{2.6}$$

où ε_+ et ε_- sont liés aux variations initiales de densité dans le champ de matière et où l'on a l'équation différentielle suivante vérifiée par δ :

$$\boxed{\ddot{\delta}+2\frac{\dot{a}}{a}\dot{\delta}-4\pi G\bar{\rho}\delta=0} \tag{2.7}$$

La dépendance en H de D_+ et D_- traduit une dépendance de ces deux grandeurs à la quantité et à la nature de l'énergie noire ainsi qu'aux autres composantes énergétiques de l'Univers.

Approximation de Zel'dovich

L'approche linéaire décrite ci-dessus n'est valide que lorsque $\delta \ll 1$. Aussi, l'approximation de Zel'dovich[57, 58] propose un traitement lagrangien des perturbations initiales du champ de densité pour dépasser un peu cette limitation. Soit $\vec{x}=\frac{\vec{r}}{a}$ les coordonnées comobiles et $\vec{r}$ les coordonnées physiques, et soit $\vec{q}$ les coordonnées lagrangiennes initiales à $t\to 0$ des particules, on peut écrire :

$$\vec{x}\left(\vec{q},D_+\left(t\right)\right)=\vec{q}+D_+\left(t\right)\vec{f}\left(\vec{q}\right)\qquad\text{avec}\qquad\vec{f}\left(\vec{q}\right)=\vec{\nabla}\Psi\left(\vec{q}\right) \tag{2.8}$$

où $\vec{f}$ est le champ de déplacement que l'on peut réécrire en terme d'un potentiel Ψ. À partir de la conservation de la masse, il vient :

$$\rho\left(\vec{x},t\right)\mathrm{d}\vec{x}=\bar{\rho}\left(t\right)\mathrm{d}\vec{q}\qquad\Rightarrow\qquad\rho\left(\vec{x},t\right)=\frac{\bar{\rho}\left(t\right)}{J\left[\frac{\partial\vec{x}}{\partial\vec{q}}\right]} \tag{2.9}$$

où $J\left[\frac{\partial\vec{x}}{\partial\vec{q}}\right]$ est le jacobien de la transformation entre les coordonnées eulériennes et lagrangiennes, et l'on peut finalement écrire :

$$\rho\left(\vec{q},t\right)=\frac{\bar{\rho}\left(t\right)}{\left[1-D_+\left(t\right)\alpha\left(\vec{q}\right)\right]\left[1-D_+\left(t\right)\beta\left(\vec{q}\right)\right]\left[1-D_+\left(t\right)\gamma\left(\vec{q}\right)\right]} \tag{2.10}$$

où $\alpha\left(\vec{q}\right)\geq\beta\left(\vec{q}\right)\geq\gamma\left(\vec{q}\right)$ sont les valeurs propres du tenseur de déformation $D_{ij}=-\frac{\partial f_i}{\partial q_j}$.

Sortie du domaine linéaire

Ces descriptions eulérienne et lagrangienne du régime linéaire des perturbations permettent de traiter l'évolution du champ de matière initial, l'approximation de Zel'dovich restant valable bien plus longtemps que le traitement eulérien. Ainsi, peu à peu, les sous-densités et les surdensités δ vont être amplifiées, à mesure que le facteur d'échelle augmente. La paramètre cosmologique σ_8 est lié à la variance des surdensités dans des sphères de rayon[3] $r_8 = 8\,\mathrm{Mpc/h}$:

$$\boxed{\sigma_8^2 = \left\langle \left(\frac{3}{4\pi r_8} \int_{\|\vec{x}\| \leq r_8} \delta(\vec{x})\, \mathrm{d}^3\vec{x} \right)^2 \right\rangle = \int_0^{\frac{2\pi}{r_8}} 4\pi k^3 P(k) \frac{\mathrm{d}k}{k} \sim 1} \tag{2.11}$$

si bien que l'entrée dans le régime non-linéaire peut être caractérisée par l'échelle r_8 qui marque l'échelle à partir de laquelle le contraste de densité devient trop grand, rompant, de fait, la validité des approximations dans la description eulérienne précédemment décrites.

2.1.3 Régime non-linéaire et scénario hiérarchique de formation

Physique gravitationnelle et physique baryonique

À partir du moment où $\delta \sim 1$, les fluctuations de densité vont être telles que leur dynamique, décrite par les équations 2.5, va entrer dans un régime d'évolution non-linéaire. C'est dans ce contexte que la formation des grandes structures que l'on peut aujourd'hui observer va s'amorcer. Sous l'effet de la gravité, les surdensités vont s'effondrer sur elles-mêmes puis s'attirer et fusionner pour former des structures de plus en plus massives. Cette évolution des petites structures vers les grandes par fusions successives présentée en figure 16 est à la base du scénario hiérarchique de formation qui va progressivement conduire à la constitution des superamas de galaxies et de la structure filamentaire. Dans cette période, deux physiques assez différentes peuvent être distinguées : la physique de la matière noire et la physique des baryons. Ou plutôt devrait-on dire la physique purement gravitationnelle non-collisionnelle et la physique du gaz, collisionnelle, puisque c'est de cela dont il s'agit. Aux très grandes échelles, la dynamique des structures peut être étudiée en négligeant les processus thermiques et en se concentrant exclusivement sur les processus gravitationnels. Mais lorsqu'il s'agit d'être en mesure de correctement décrire la physique interne des amas ou l'évolution des galaxies, cette approximation n'est plus valable et la physique du gaz, beaucoup plus complexe, doit être prise en compte.

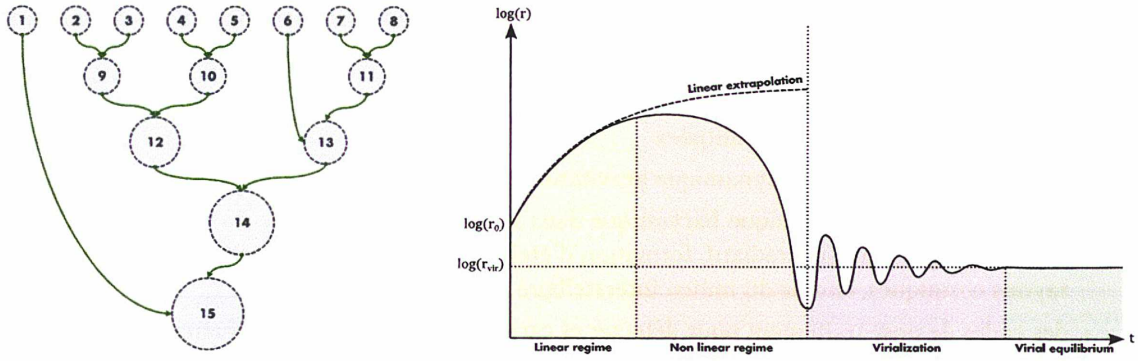
Effondrement d'un halo sphérique

Dans le cas purement gravitationnel, le traitement analytique de l'effondrement d'un halo sphérique dans un univers Einstein-de-Sitter, c'est-à-dire sans courbure et dominé par la matière, permet d'établir certains repères quant à l'évolution non-linéaire des surdensités initiales. Pour cela on isole une région sphérique surdense de rayon r et de densité $\rho = \rho_c + \delta\rho$. Cette région peut alors être traitée comme un univers de courbure positive avec $\Omega_M = 1 + \delta$. Or l'évolution d'un univers de Friedmann-Lemaître dans ces conditions peut s'écrire sous la forme paramétrique :

$$\left.\begin{aligned} r(\theta) &= A(1 - \cos\theta) \\ t(\theta) &= B(\theta - \sin\theta) \end{aligned}\right\} \quad \text{avec} \quad \left\{\begin{aligned} \theta &= H_0 \eta \sqrt{\Omega_m - 1} \\ A &= \frac{r_0 \Omega_{M_0}}{2(\Omega_{M_0} - 1)} \\ B &= \frac{\Omega_{M_0}}{2H_0 (\Omega_{M_0} - 1)^{\frac{3}{2}}} \end{aligned}\right. \tag{2.12}$$

avec un rayon maximal $r_{max} = 2A$ atteint en $\theta_{max} = \pi$, $t_{max} = B\pi$ et un effondrement complet $r \to 0$ en $\theta_c = 2\pi$, $t_c = 2\pi B$. La valeur de la surdensité au moment où le rayon est à son maximum et commence à décroître est :

$$1 + \delta_{max} = \frac{\rho_{max}}{\rho_c} = \frac{\Omega_{M_0}\rho_{c_0}}{r_{max}^3} \frac{a^3}{\rho_{c_0}} = \Omega_{M_0} \frac{\left(\frac{3}{2} H_0 t_{max}\right)^2}{r_{max}^3} = \frac{9\pi^2}{16} \approx 5.55 \tag{2.13}$$



(a) Fusions successives de structures.

(b) Effondrement d'une sphère surdense[60].

FIGURE 16 – Formation de structures dans le régime non-linéaire. À gauche, le principe du scénario hiérarchique de formation est illustré : à partir des surdensités initiales, des structures de plus en plus massives sont formées par fusions successives. Ainsi, dans cet exemple, la structure 15 observée à l'heure actuelle a pour progéniteurs les halos 1 et 14, ce dernier ayant lui-même pour progéniteurs les structures 12 et 13 *etc.* En procédant de cette façon, on peut remonter aux surdensités initiales qui ont donné naissance aux amas actuels. Si l'on s'intéresse à l'évolution de chacune de ces structures de façon isolée et dans un cas sphérique, celle-ci se déroule en quatre phases : une phase d'évolution linéaire, une phase d'effondrement non-linéaire, suivie d'une phase de virialisation pour finir à l'équilibre viriel, une fois la structure stabilisée.

On peut également montrer par un développement de la forme paramétrique 2.12 que l'effondrement complet est atteint lorsque la surdensité prédite par la théorie linéaire vaut[59] :

$$\delta_c = \frac{3}{20}\left(\frac{6\pi \times 2t_{max}}{t_{max}}\right)^{\frac{2}{3}} \approx 1.686 \tag{2.14}$$

Dans la réalité, la région surdense ne s'effondre pas jusqu'à un rayon nul : la structure se stabilise à l'équilibre viriel au moment où le rayon de la sphère est égal à la moitié de son rayon maximal $r_{vir} = \frac{r_{max}}{2}$, lorsque $t_{vir} = 2t_{max}$. À ce moment-là, la densité de la sphère a été multipliée par $2^3 = 8$, tandis que, comme $a \propto t^{\frac{2}{3}}$, celle de l'Univers a été divisée par $2^2 = 4$ si bien que :

$$1 + \delta_{vir} = (1 + \delta_{max}) \times 2^3 \times 2^2 = 18\pi^2 \approx 178 \tag{2.15}$$

Dans la pratique, une surdensité $\delta_{vir} = 200$ associée à un rayon r_{200} est souvent utilisée pour définir des structures virialisées.

L'effondrement d'un halo parfaitement sphérique dans un univers Einstein-de-Sitter est bien entendu une approximation : la réalité est plus complexe, l'expansion est accélérée et les structures ne sont pas sphériques. Dès lors, l'approche analytique se retrouve vite limitée et ne suffit plus pour décrire la dynamique de ces processus non-linéaires. Aussi, le recours à la simulation numérique va permettre de mieux comprendre la formation des grandes structures dans un univers en expansion et d'en fournir une caractérisation, là où l'approche analytique ne pouvait que donner accès à des ordres de grandeur.

2.2 Simuler l'évolution des grandes structures

2.2.1 Fondements et principes des simulations cosmologiques

Déroulement d'une simulation

Pour explorer ce qui sépare le scénario décrit en figure 16 des résultats des grands relevés observationnels de la figure 13, la cosmologie va s'appuyer sur le calcul haute performance. En effet, parvenir à suivre l'évolution d'une structure filamentaire sur des dizaines, centaines ou milliers de mégaparsecs

demande des ressources de calcul et de stockage non négligeables. Dans cet objectif, les simulations cosmologiques reposent généralement sur cinq composantes distinctes :

- un intégrateur de la cosmologie homogène
- un générateur de conditions initiales
- un solveur N-corps de la dynamique gravitationnelle
- des solveurs pour la physique baryonique dans le cas où la dynamique du gaz est traitée (hydrodynamique, transfert radiatif, formation d'étoiles, rétroactions, production et évolutions des rayons cosmiques, chimie du milieu interstellaire...)
- des codes de post-traitement pour détecter et caractériser les halos et les champs

Modèle homogène et conditions initiales

Avant de commencer à suivre l'effondrement de la matière, deux éléments sont nécessaires. Le premier est la solution $a(t)$ du modèle friedmannien au sein duquel la structuration sera suivie. Il s'agit là d'un élément capital : métrique homogène et contenu matériel sont traités indépendamment, l'évolution du second ne pouvant donc pas rétroagir sur la dynamique de la première. On étudie donc la structuration au sein d'un fond de métrique homogène. Le second élément nécessaire a trait aux conditions initiales, c'est-à-dire aux sous-densités et aux surdensités originelles. À part dans le cas très particulier de simulations contraintes, on ne cherche pas à reproduire l'Univers tel qu'il est réellement : on cherche à reproduire l'Univers tel qu'il est en termes statistiques. La compatibilité aux observations repose ainsi sur les propriétés statistiques de la distribution des structures. Les perturbations initiales du champ de matière sont générées pour être compatible avec le spectre de puissance du fond diffus cosmologique. Une infinité de distributions étant possible, chaque simulation est caractérisée par le tirage aléatoire dont elle est issue, c'est-à-dire sa phase. Aussi, lorsqu'il s'agit de comparer la structuration pour deux cosmologies différentes, c'est-à-dire pour deux évolutions $a(t)$ différentes, on s'assure que les deux simulations reposent sur la même phase de façon à ce que les différences ne puissent venir que de la dynamique. Une fois la phase sélectionnée et la distribution à $z \approx 1100$ générée, cette dernière est évoluée linéairement jusqu'à un redshift typiquement compris entre 200 et 50. Cela permet, entre autres, d'éviter de commencer à accumuler des erreurs numériques dans un régime parfaitement décrit par une approche analytique.

Évolution dynamique

La suite de l'évolution du champ de matière est prise en charge par les codes de simulation dynamiques. Dans un cadre cosmologique, des conditions au bord périodiques sont choisies : il s'agit d'étudier la structuration dans un tore tridimensionnel. C'est donc le champ de matière au sein d'une boîte cubique périodique en coordonnées comobiles [1] dont on suit l'évolution. Pour se faire, la physique gravitationnelle et la physique du gaz sont traitées différemment. Pour la première, le champ de matière est discrétisé en termes de particules, le problème de l'évolution devenant alors un problème de dynamique gravitationnelle à N-corps, avec N actuellement typiquement compris entre 10^6 et 10^{12}, et pour lequel on cherche à résoudre le système, en coordonnées supercomobiles :

$$\boxed{\begin{aligned} \frac{\mathrm{d}\vec{x}}{\mathrm{d}t} &= \vec{v} \\ \frac{\mathrm{d}\vec{v}}{\mathrm{d}t} &= -\vec{\nabla}\Phi \\ \nabla^2\Phi &= 4\pi G\rho \end{aligned}} \tag{2.16}$$

Dans le cas où la seule dynamique gravitationnelle est résolue, on parle de simulation purement N-corps, ou de simulation de matière noire pure car le problème traité est de nature non-collisionnelle.

1. Ou supercomobiles[61].

Pour le second type de physique, la physique baryonique, qui joue un rôle essentiel à l'échelle galactique et à l'échelle des amas, plusieurs descriptions sont communément employées : des descriptions en termes de particules, donnant lieu à un traitement SPH (Smoothed-Particle Hydrodynamics), ou des descriptions fluides en termes de volumes finis basées sur des maillages cubiques, adaptatifs ou non, ou sur des tessellations de l'espace de simulation. En se basant sur ces approches, la dynamique newtonienne non-linéaire de la distribution originelle de matière est donc suivie au sein d'un univers en expansion, pas de temps après pas de temps. Dans un contexte cosmologique, les boîtes de simulation sont choisies d'une taille supérieure à l'échelle d'homogénéité de façon à ce que le volume simulé soit représentatif de l'Univers. Des tailles de plusieurs centaines de mégaparsecs sont ainsi fréquemment utilisées. Lorsque l'hydrodynamique doit être prise en compte, des volumes sensiblement plus petits, de l'ordre de quelques dizaines de mégaparsecs, sont utilisés du fait la nécessité de résoudre de plus petites échelles. Pour cette raison, il est difficile d'étudier conjointement l'effet de la cosmologie et l'effet de la physique du gaz sur la formation des grandes structures. Toutefois, l'hypothèse ergodique, si elle était vérifiée, permettrait de sonder l'effet de la cosmologie non pas en simulant des grands volumes, mais en simulant l'évolution d'un très grand nombre de réalisations de taille plus modeste. Il s'agit là d'une hypothèse souvent utilisée mais dont la validité demande à être testée du fait des couplages d'échelles qui interviennent dans le régime non-linéaire : en effet, dès lors que l'on sort du domaine linéaire, la structuration aux grandes échelles rétroagit sur les petites échelles et *vice versa*. Or le fait de simuler des volumes plus petits que l'échelle d'homogénéité va nécessairement couper ce genre de rétroactions. Ce type d'effets peut être étudié en comparant les résultats statistiques au sein d'une simulation de grand volume et au sein du même volume total simulé au moyen d'un grand nombre de simulations indépendantes de volume plus réduit.

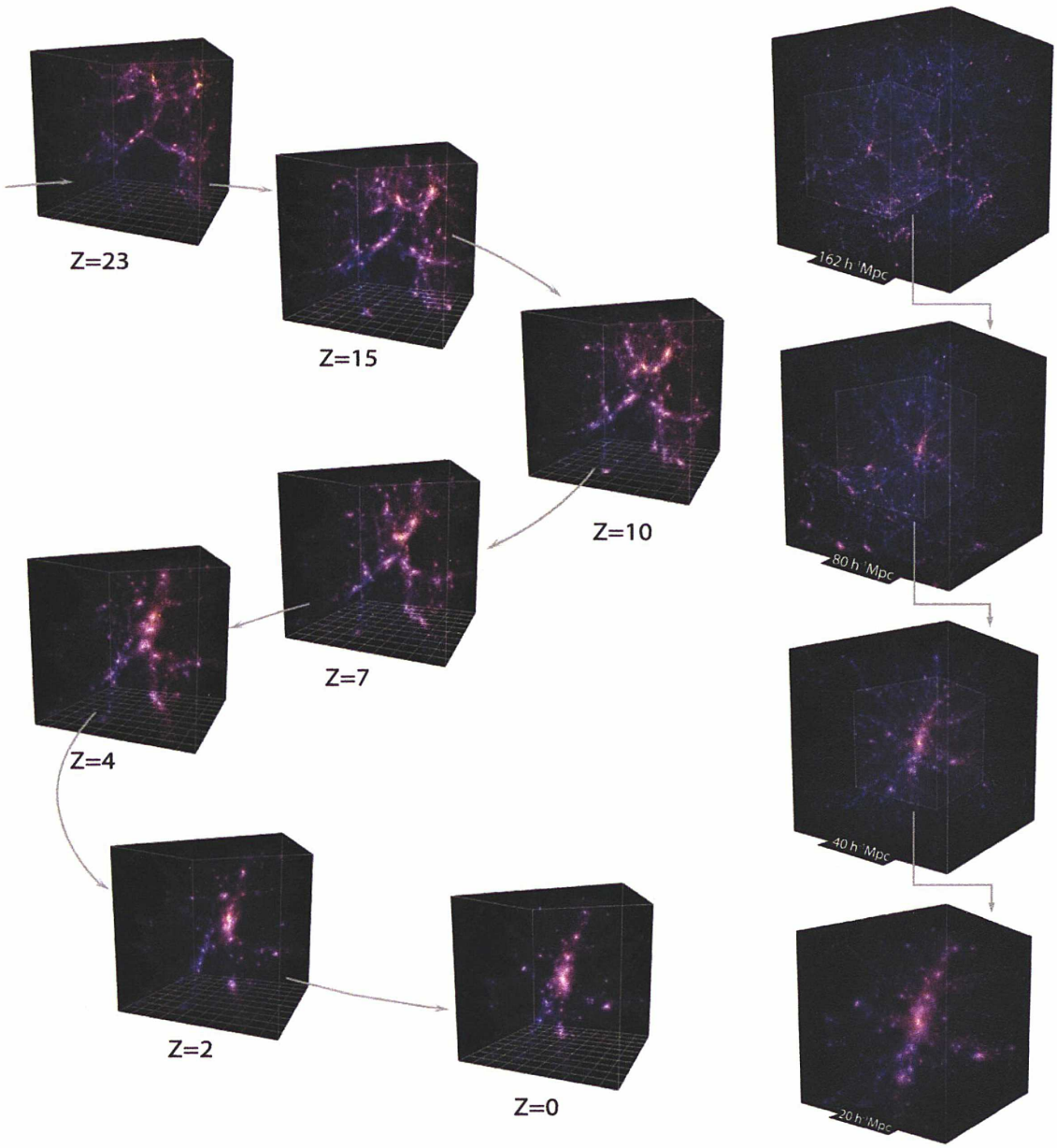
Découpage du champ de matière et détection des halos

À l'issue d'une simulation cosmologique, on obtient donc l'état du champ de matière à différents redshifts comme illustré en figure 17. Toutefois, ce champ de matière n'est encore qu'à l'état de particules et de quantités hydrodynamiques le cas échéant. Extraire de l'information physique de ces simulations demande de réaliser des analyses, qu'elles soient faites à la volée par le code dynamique lui-même ou qu'elles soient faites en post-traitement, comme c'est plus souvent le cas. L'un des points clefs de ce processus réside dans la détection de structures. Et cela est loin d'être simple : de multiples algorithmes de détection existent, sur des critères de surdensité, sur des critères de percolation ou encore sur des critères de virialisation, et de multiples définitions de ce qu'est une structure existent également, tant au niveau théorique qu'au niveau observationnel. Une autre approche consiste, plutôt qu'à détecter les surdensités, à détecter les vides. Mais, là encore, de nombreuses méthodes existent. Quoi qu'il en soit, la fin du calcul dynamique ne marque que le commencement des problématiques liées à la caractérisation de la structure filamentaire de l'Univers. Car, comme on peut le voir en figure 17, c'est bien une structure comparable à celle révélée par les observations et présentée en figure 13 qui est obtenue.

2.2.2 Caractérisation des champs et des halos de matière noire

Profil des halos

L'étude des halos de matière noire joue un rôle central dans l'analyse de la structuration, car ils en constituent les éléments de base à grande échelle. Leur distribution, leurs caractéristiques morphologiques, leurs caractéristiques physiques sont autant d'éléments susceptibles de dépendre de l'histoire de l'effondrement, et par conséquent de l'histoire du facteur d'échelle. À partir des simulations il est possible d'en définir un profil typique. Le profil de Navarro–Frenk–White[62], abrégé NFW, fait partie



(a) Évolution d'une simulation numérique du projet DEUS.

(b) Agrandissement d'une zone structurée.

FIGURE 17 – Simulation de formation de structures. Les simulations numériques permettent de suivre la dynamique non-linéaire du champ de matière à partir des surdensités initiales observées au moment de la recombinaison. Elles consistent à faire évoluer une boîte comobile d'Univers avec des conditions au bord périodiques de facteur d'échelle en facteur d'échelle comme illustré à gauche. Au début, la distribution de matière est quasi-homogène, puis celle-ci s'effondre de plus en plus pour former des halos et des filaments de plus en plus grands et de plus en plus massifs. Sur les images, l'intensité lumineuse correspond à la densité de particules : les zones jaunes correspondent aux cœurs de halos tandis que les zones violettes sont plus diffuses. Dans un contexte cosmologique, il est préférable de prendre des volumes de simulation plus grands que l'échelle d'homogénéité de façon à ce que la boîte puisse être statistiquement représentatif d'un volume d'Univers. À gauche, on agrandit progressivement une zone de 20 Mpc/h au sein d'une boîte de 162 Mpc/h à $z = 0$. Ce qui, à grande échelle, apparaissait comme relativement homogène se révèle alors être une structure filamentaire complexe à l'image de ce qui peut être observé par les grands relevés, comme illustré en figure 13. Il s'agit, pour ces deux exemples, de simulations purement gravitationnelles, c'est-à-dire sans physique du gaz, dont le rôle peut être important à plus petite échelle.

des modèles les plus utilisés. Celui-ci repose sur un profil de densité :

$$\boxed{\rho(r) = \frac{\rho_s}{\frac{r}{r_s}\left(1+\frac{r}{r_s}\right)^2}} \tag{2.17}$$

où r est la distance au centre du halo, et où ρ_s et r_s sont les deux paramètres du modèle, qui varient d'un halo à l'autre. Dans le cadre de ce formalisme, on pose de plus le paramètre de concentration $c = \frac{r_{vir}}{r_s}$ où le rayon viriel est souvent défini comme le rayon à l'intérieur duquel la densité moyenne est égale à 200 fois la densité critique comme précisé en 2.15 : $r_{vir} = r_{200}$. La masse du halo M_{200} associée vaut alors :

$$M_{200} = \frac{4}{3}\pi 200 \rho_c r_{200}^3 \tag{2.18}$$

tandis que la masse intégrée jusqu'à un rayon r d'un halo NFW vaut :

$$M(r) = \int_0^r 4\pi r'^2 \rho(r')\,\mathrm{d}r' = 4\pi\rho_s r_s^3 \left[\ln\left(\frac{r_s+r}{r_s}\right) - \frac{r}{r_s+r}\right] \tag{2.19}$$

Ce genre de modèle de halo sphérique se couple bien avec des algorithmes de détection comme Spherical Overdensity[63] (SO) qui font progressivement grandir des sphères autour des pics de densité obtenus dans les simulations et s'arrêtent sur un critère de densité. Mais lorsque les halos sont détectés par percolation avec des algorithmes comme Friends-of-Friends[64, 65, 66] (FoF), les profils peuvent devenir beaucoup plus complexes comme illustré en figure 18. De tels profils dépendent fortement de l'histoire de formation des halos et des fusions successives dans le cadre du modèle hiérarchique. Si la dépendance des profils rencontrés est forte en fonction des algorithmes considérés, elle est encore pire lorsqu'il s'agit de s'intéresser aux sous-halos[67]. Car en effet, pour chaque structure considérée sur un critère X, il est en général possible d'isoler des sous-structures à l'intérieur sur un critère plus restrictif X'. La définition rigoureuse de ce qu'est une « structure » est donc un problème complexe. Cependant, dans un premiers temps, l'étude de l'effet de la cosmologie sur la formation des structures peut faire abstraction de ces raffinements.

Fonctions de masse analytique et numérique

L'une des quantités importantes les plus simples pour l'étude de la structuration est la fonction de masse. Celle-ci correspond à la fraction de halos par unité de volume n dont la masse est comprise entre M et $M + \mathrm{d}M$:

$$\boxed{F = \frac{\mathrm{d}n}{\mathrm{d}M} = f(\sigma)\frac{\bar{\rho}}{M}\frac{\mathrm{d}\ln\sigma^{-1}}{\mathrm{d}\ln M}} \text{ avec } \begin{cases} \sigma^2(M,z) = \dfrac{b^2(z)}{2\pi^2}\displaystyle\int_0^\infty k^2 P(k) W^2(k,M)\,\mathrm{d}k \\ f_{PS}(\sigma) = \sqrt{\dfrac{2}{\pi}}\dfrac{\delta_c}{\sigma}e^{-\frac{\delta_c^2}{2\sigma^2}} \\ f_{ST}(\sigma) = A\sqrt{\dfrac{2a}{\pi}}\left[1+\left(\dfrac{\sigma^2}{a\delta_c^2}\right)^p\right]^2 \dfrac{\delta_c}{\sigma}e^{-\frac{a\delta_c^2}{2\sigma^2}} \\ f_J(\sigma) = \alpha e^{-|\ln\sigma^{-1}+\beta|^\gamma} \end{cases} \tag{2.20}$$

où :

- $\bar{\rho}$ est la densité moyenne de l'Univers au redshift considéré
- $b(z) = \frac{D_+(z)}{D_+(z=0)}$ est le facteur de croissance normalisé à 1 en $z = 0$
- $W(k, M(r)) = 3\frac{\sin(kr)-kr\cos(kr)}{(kr)^3}$ est la représentation dans l'espace de Fourier d'un filtre « top-hat »
- δ_c est la surdensité extrapolée linéairement au moment de l'effondrement définie en 2.14

- $f(\sigma)$ est la fonction de multiplicité, qui permet d'isoler les termes de la fonction de masse qui ne dépendent en principe que des problématiques d'effondrement et dont la spécification semi-analytique est à même de donner une prédiction de la fonction de masse
- f_{PS} est la fonction de multiplicité de Press et Schechter[68]
- f_{ST} est la fonction de multiplicité de Sheth et Tormen[69] avec $A = 0.3222$, $a = 0.707$ et $p = 0.3$
- f_J est la fonction de multiplicité de Jenkins[70] obtenue à partir de simulations numériques dans le domaine $-1.2 \leq \ln \sigma^{-1} \leq 1.05$ avec $\alpha = 0.315$, $\beta = 0.61$, $\gamma = 3.8$

Les quelques formes fonctionnelles indiquées correspondent à celles qui sont utilisées le plus abondamment dans la littérature pour prédire la fonction de masse. La fonction de multiplicité, elle, permet d'étudier l'effet de la cosmologie sur le processus d'effondrement.

Dans le cas de simulations numériques, le calcul de la fonction de masse est beaucoup plus direct, puisqu'il suffit d'établir un histogramme de la fraction de halos détectés N_{halos} au sein du volume de simulation V_{box} dont la masse est comprise entre M_{halo} et $M_{\text{halo}} + \mathrm{d}M_{\text{halo}}$:

$$F(M) = \frac{1}{V_{\text{box}}} \frac{\mathrm{d}N_{\text{halos}}}{\mathrm{d}M_{\text{halo}}} \tag{2.21}$$

Il est donc, de fait, beaucoup moins sujet à d'éventuelles approximations. Les deux principales dépendances non-physiques de cette fonction de masse numérique sont d'une part l'algorithme de détection utilisé et d'autre part les masses minimale et maximale des halos formés. La masse minimale dépend de la résolution en masse de la simulation, qui introduit un effet de seuil : pour être à même de résoudre la dynamique d'un halo, il faut, dans l'idéal, que celui-ci soit typiquement composé d'au moins une centaine de particules. Il est possible d'abaisser ce seuil de détection mais l'on devient alors davantage dépendant de l'algorithme utilisé. La masse maximale, elle, dépend essentiellement de la taille de boîte : la fonction de masse chute fortement à très grande masse et il est nécessaire de disposer d'un très grand volume de simulation pour être à même de sonder statistiquement cette zone. De plus, c'est dans ce type de situation de sondage d'évènements rares que l'hypothèse ergodique est susceptible de ne plus être vérifiée : former des halos de $10^{16}\,\mathrm{M}_\odot/\mathrm{h}$ au sein de N boîtes de volume restreint V est plus difficile qu'au sein d'une seule boîte de taille NV. Sonder la queue de la distribution demande donc de réaliser des simulations d'un volume suffisant, indépendamment de leur nombre.

2.2.3 Note sur le rôle des baryons

La physique gravitationnelle ne suffit pas à produire l'Univers que l'on observe : les étoiles, les nébuleuses, les galaxies sont autant de produits de la dynamique du gaz et des processus thermiques qui l'accompagnent. Au moment de la recombinaison les contrastes de densité du gaz et de la matière noire sont à peu près égaux $\delta_G \approx \delta_{DM}$, mais tandis que les seconds vont croître linéairement en fonction de D_+, les modes des premiers correspondants à la masse de Jeans M_J vont voir leur taux de croissance atténué du fait de la pression :

$$\boxed{M_J = \frac{4\pi}{3}\bar{\rho}\left(\frac{2\pi a}{k_J}\right)^3} \qquad \text{avec} \qquad k_J = \frac{a\sqrt{4\pi G\bar{\rho}}}{c_s} \tag{2.22}$$

où c_s est la vitesse du son. En conséquence, le gaz va naturellement tomber dans les halos dont la masse est supérieure à la masse de Jeans, alors qu'il va rester diffus autour de ceux dont la masse est plus petite. Il existe ainsi un effet de seuil entre les halos qui vont former des étoiles et ceux qui n'en formeront pas.

La modélisation de ces processus et de tous les échanges thermiques intra-halo et inter-halos est nettement plus complexe que le traitement de la physique gravitationnelle pure du fait de la variété des phénomènes en jeu ne serait-ce que parce qu'il faut au moins y inclure d'une part la formation d'étoiles et d'autre part le feedback afin d'éviter le problème de l'overcooling qui conduit le gaz à

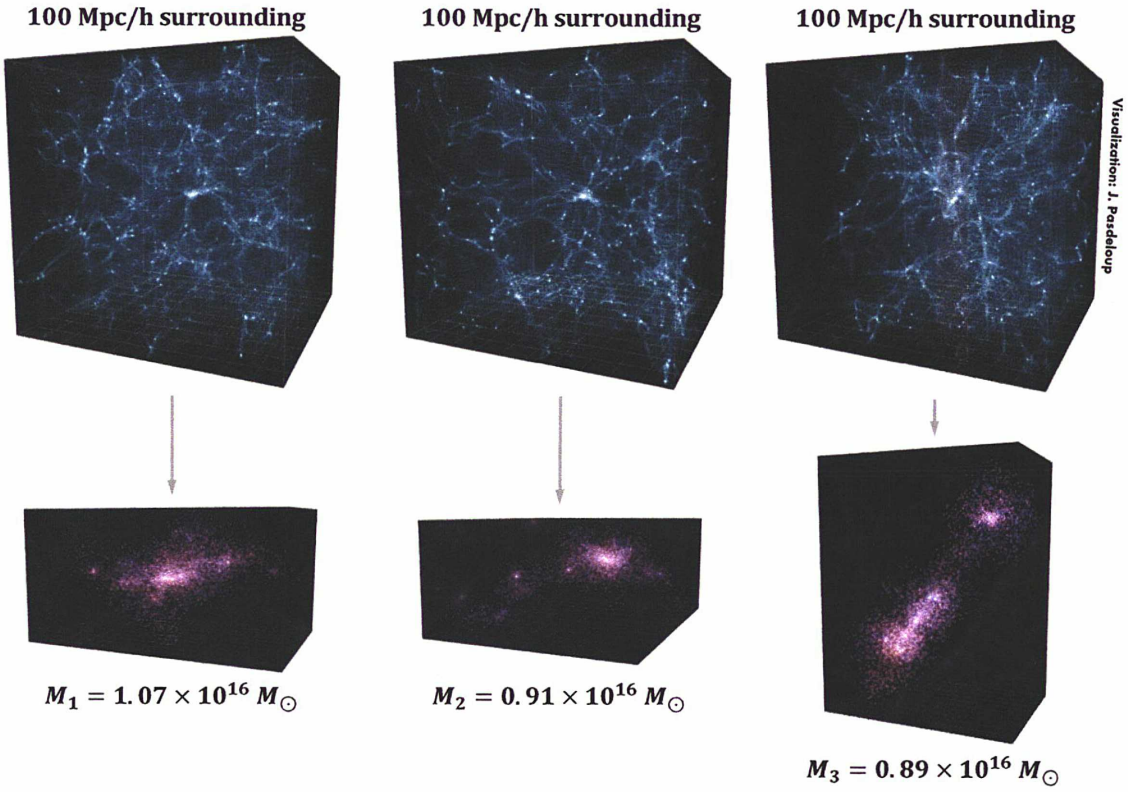


FIGURE 18 – Halos les plus massifs de la simulation DEUS-FUR ΛCDM à $z = 0$. Des algorithmes de percolation, comme Friends-of-Friends permettent de détecter des halos avec des profils parfois assez éloignés du cas sphérique idéal, comme par exemple celui de droite ci-dessus. Chaque point violet représente une particule de la simulation. Ces trois halos d'environ $10^{16}\,\mathrm{M}_\odot$ ont été obtenus dans le cadre du projet Full Universe Run au sein d'une boîte de 21 $\mathrm{Gpc/h}$ de côté, soit la taille de l'Univers Observable. Sonder la limite supérieure de la fonction de masse demande des simulations de très grand volume du fait de la rareté de ce type d'évènements.

s'effondrer trop rapidement au cœur des halos. Les modèles étant à la fois multi-paramétriques et fortement dégénérés vis-à-vis des observations cela conduit souvent à réaliser des simulations basées sur des prescriptions empiriques et des modèles sous-grilles. Aussi, dans ce contexte, si comprendre la physique des amas et arriver à reproduire l'évolution du taux de formation d'étoiles dans l'Univers sont des questions importantes, elles peuvent être, dans un premier temps, dissociées des problématiques cosmologiques à grande échelle avec lesquelles elles entretiennent de toute façon des dégénérescences.

Dans la suite, on se placera donc exclusivement dans le cadre de simulations sans physique baryonique. À première vue, chercher à analyser des effets tels que la propagation de la lumière dans des univers sans matière lumineuse peut sembler paradoxal. Mais de tels effets sont avant tout dictés par la métrique et la structuration à grande échelle et c'est pourquoi la physique baryonique peut être négligée. Car, dans un premier temps, le but n'est ni de s'intéresser aux interactions du rayonnement et du gaz dans les halos, ni de produire des catalogues virtuels de sources affectées par les effets liés à la propagation de la lumière dans un univers structuré : il s'agit déjà de comprendre la nature de ces effets, leur amplitude, et leur dépendance à la cosmologie.

2.3 Sonder la cosmologie au moyen de la structuration

2.3.1 Empreinte de la cosmologie sur la formation des structures avec une statistique maximale

Empreinte de la cosmologie homogène sur l'effondrement

Comme nous l'avons vu dans le premier chapitre, plusieurs modèles cosmologiques possédant des histoires de taux d'expansion légèrement différentes sont à même de reproduire les observations actuelles, au niveau homogène (supernovæ) ou linéaire (fond diffus cosmologique). Or, si l'on voit les inhomogénéités comme autant de petites perturbations à un fond de métrique homogène, ces variations de $a(t)$ vont entraîner des histoires d'effondrement telle que celle présentée en figure 16 différentes. Ces différences vont être amplifiées par le caractère non-linéaire de la dynamique ce qui va finalement mener à des modifications des fusions successives des halos. La cosmologie homogène va ainsi laisser son empreinte dans la structuration, aux très grandes échelles, comme à l'échelle des halos[35, 71, 72]. Ces structures sont donc une sonde cosmologique capable de briser les dégénérescences entre les modèles. De plus, c'est justement de cette structuration que va dépendre la propagation de la lumière. Avant d'étudier la seconde en détail, il est donc nécessaire de s'intéresser à la première, au moins dans les grandes lignes. C'est l'objet des paragraphes ci-après.

Le Full Universe Run et la structuration à l'échelle de l'Univers Observable

Trois sondes de la structuration vont être étudiées : le spectre de puissance, la fonction de masse et les oscillations acoustiques des baryons. Pour cela, nous allons nous appuyer sur les trois simulations principales du projet Dark Energy Universe Simulation : Full Universe Run (abrégé DEUS-FUR dans la suite). Elles sont basées sur le modèle de concordance (ΛCDMW7), sur un modèle de quintessence de Ratra-Peebles (RPCDMW7) et sur un modèle d'énergie noire phantom (WCDMW7) et ont été calibrées par rapport aux données de WMAP après sept ans d'observations. Les paramètres cosmologiques associés à ces modèles ont été donnés dans le premier chapitre. Il s'agit des premières simulations de la structuration sur le volume de l'Univers Observable, avec une taille de boîte de 21 Gpc/h de côté, $8192^3 \approx 550 \times 10^9$ particules en interaction gravitationnelle et une résolution en masse d'environ $10^{12}\,\mathrm{M}_\odot/\mathrm{h}$. Davantage de détails concernant leur réalisation en 2012 sur le supercalculateur Curie sont donnés dans la partie numérique de ce manuscrit. Bien entendu, une telle résolution en masse ne permet pas de sonder finement la dynamique des amas : étant donné qu'au moins une centaine de particules est nécessaire pour correctement décrire un halo, cela signifie que ces simulations sont à même de sonder efficacement les structures de plus de $10^{14}\,\mathrm{M}_\odot/\mathrm{h}$. Mais disposer de la structuration sur le volume de l'Univers Observable est un atout considérable. En effet cela permet, entre autres choses, d'accéder à une statistique maximale sur les propriétés de l'Univers à très grande échelle, de sonder des évènements les plus rares, et de construire, pour la première fois, des espaces de redshift allant potentiellement jusqu'au fond diffus cosmologique ce qui est essentiel pour l'étude de la propagation de la lumière dans tout l'Univers Observable.

2.3.2 Propriétés du champ de matière : l'exemple du spectre de puissance

Tracer le spectre de puissance des fluctuations du champ de matière est une façon simple et efficace de caractériser la structuration. La figure 19 présente l'évolution de tels spectres de $z = 100$ à $z = 0$ pour les trois cosmologies du Full Universe Run. Une simulation est caractérisée par deux échelles : sa taille de boîte L_{box} et sa résolution spatiale L_{res}. La première contraint les grandes échelles qu'il est possible d'explorer, donc les petites valeurs de $k = \frac{2\pi}{L}$ tandis que de la seconde fixe la limite d'exploration des petites échelles, c'est-à-dire des grandes valeurs de k. Dans le cas du Full Universe Run, un algorithme de maillage adaptatif permet d'améliorer la résolution dans les régions les plus denses. La répartition de la puissance entre ces deux échelles limites renseigne sur les tailles caractéristiques des grandes structures. Pour la suite, il est utile de préciser que le choix de phase était le même pour les

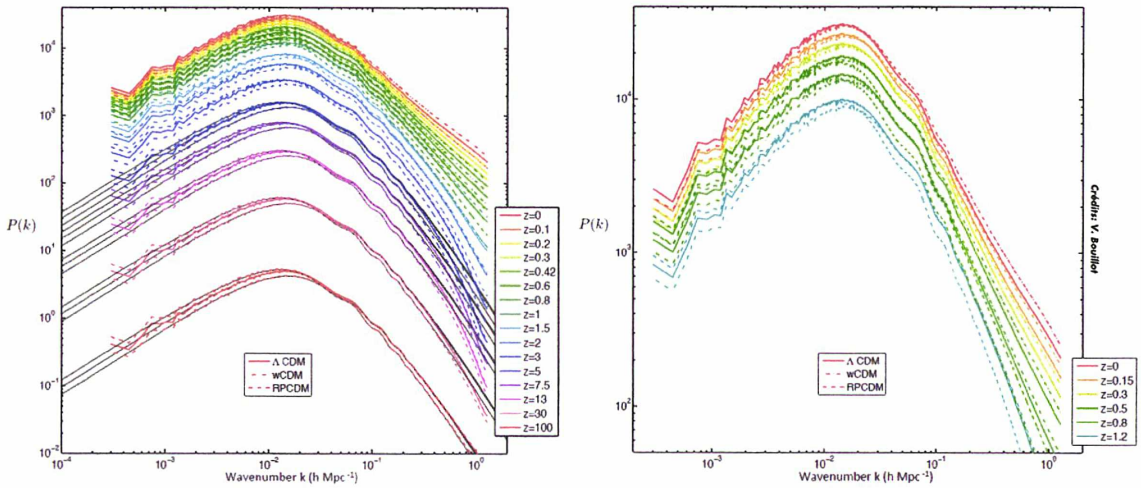


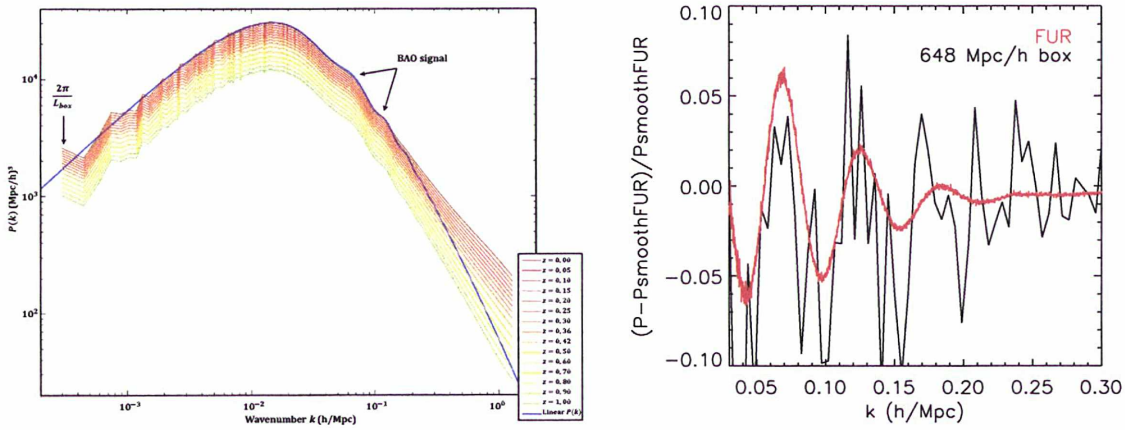
FIGURE 19 – Évolution des spectres de puissance des modèles ΛCDM, RPCDM et WCDM simulés dans le cadre du Full Universe Run[73]. Les spectres sont tracés pour différents redshifts allant de $z = 100$ à $z = 0$. La figure à droite constitue un agrandissement de celle de gauche afin de mieux exhiber le comportement des spectres à faible redshift. k_{min} est donné par la longueur des côtés des boîtes de simulation $\frac{2\pi}{L_{\text{box}}}$ tandis que k_{max} dépend de la résolution spatiale de ces dernières. Les courbes en noir à bas redshift correspondent aux spectres de puissance extrapolés linéairement à partir du taux de croissance D_+. À chaque redshift, il est possible de comparer les trois modèles : aux grandes échelles, c'est-à-dire pour les faibles valeurs de k, la puissance est supérieure dans le cas du modèle de quintessence RPCDM comparativement aux deux autres modèles, tandis qu'aux petites échelles, pour les valeurs de k élevées, c'est pour le modèle phantom WCDM qu'elle est la plus grande. La puissance se répartie donc assez différemment en fonction du modèle d'énergie noire. Les oscillations autour de $k \sim 10^{-1}\,\text{h/Mpc}$ correspondent aux oscillations acoustiques des baryons.

trois cosmologies : les conditions initiales ont été construites à partir du même tirage aléatoire afin de faciliter les comparaisons et afin de pouvoir isoler plus facilement les effets liés à la cosmologie. Cela explique que les oscillations entraînées par la phase dans le spectre aux grandes échelles et jusqu'à environ $8 \times 10^{-3}\,\text{h/Mpc}$ soient les mêmes dans les trois cas.

Les spectres de la figure 19 révèlent déjà ce qui était attendu, à savoir que la cosmologie homogène laisse son empreinte dans la structuration : aux grandes échelles, le spectre du modèle de quintessence de Ratra-Peebles domine celui du modèle de concordance qui domine lui-même celui du modèle phantom, tandis qu'aux petites échelles, c'est le contraire, et c'est celui du modèle phantom qui domine les deux autres. Dit autrement, le modèle phantom conduit à la formation de davantage de structures et de halos que le modèle de concordance alors que le modèle de quintessence reste plus homogène. C'est en ce sens que sonder la structuration permet de sonder la cosmologie sous-jacente par l'intermédiaire de l'empreinte qu'elle y laisse. On peut constater sur la figure 19 qu'à mesure que le redshift diminue, les effets de la dynamique non-linéaire qui se produisent originellement aux petites échelles se propagent peu à peu vers les grandes, conformément au scénario hiérarchique de formation des structures. Cela atténue progressivement les oscillations observables autour de $k \sim 10^{-1}\,\text{h/Mpc}$ qui constituent le signal BAO, les oscillations acoustiques des baryons et qui font aujourd'hui l'objet de nombreux programmes d'observation[74, 75, 76, 77].

2.3.3 Les oscillations acoustiques des baryons : un échelle des distances standard ?

Les oscillations présentes sur les spectres 19 et 20 trouvent leur origine dans l'interaction rayonnement-matière au moment de la recombinaison. En effet, avant cette dernière, des ondes acoustiques pouvaient librement se propager dans le plasma formé par les baryons et les photons. Or, au moment où l'Univers est devenu transparent, ce milieu de propagation a brutalement cessé d'exister. Ces ondes acoustiques, ont, à ce moment-là, laissé une empreinte caractéristique dans les fluctuations de densité du champ



(a) Spectre de puissance du modèle de concordance.

(b) Signal des oscillations acoustiques des baryons.

FIGURE 20 – Détection des oscillations acoustiques de baryons dans la simulation du modèle de concordance du Full Universe Run. À gauche, les différences entre le spectre non-linéaire et le spectre linéaire extrapolé au moyen du facteur de croissance sont bien visibles. La puissance aux petites échelles augmente peu à peu du fait de la dynamique non-linéaire et se propage vers des échelles de plus en plus grandes conformément au modèle hiérarchique de formation des structures. Ce redressement du spectre a pour effet d'atténuer et de modifier le signal des oscillations acoustiques des baryons. Toutefois, du fait de la très grande qualité de la statistique à grande échelle du Full Universe Run, ce signal peut être extrait de la simulation comme sur la figure de gauche où $P^{\mathrm{FUR}}_{\mathrm{smooth}}(k)$ correspond au spectre sans oscillations. Quatre oscillations sont discernables : l'amplitude de la première crête à crête est de l'ordre de 13% alors que celle de la quatrième n'est que de l'ordre d'un demi pourcent. À titre de comparaison, le même signal extrait d'une simulation de 648 Mpc/h de côté est tracé.

de matière, empreinte que l'on peut aujourd'hui sonder par l'observation de la structuration à grande échelle et par la mesure du spectre de puissance associé. Ainsi, les oscillations acoustiques des baryons permettent d'apposer des contraintes sur les paramètres cosmologiques en utilisant la position des pics et la distance angulaire.

Mais, comme cela apparait sur les spectres de puissance de la partie précédente, la forme du signal BAO est affecté par la dynamique non-linéaire. Aussi, la non prise en compte de ces effets peut induire des biais sur l'estimation des paramètres cosmologiques[78]. Dans ce contexte, les simulations numériques de formation des structures apparaissent comme un outil adapté à la quantification de ces phénomènes. Toutefois, un problème important vient du fait que l'empreinte des BAO est une empreinte laissée à grande échelle, typiquement de l'ordre de 10^2 Mpc/h, mais dont la forme provient d'échelles plus petites, dans la gamme du mégaparsec à la dizaine de mégaparsecs. Pour cette raison, accéder à des prédictions du signal des oscillations acoustiques des baryons précises au pourcent près est une tâche complexe. Cependant, parce qu'elles permettent de disposer d'une part d'une très grande statistique à l'échelle des BAO et d'autre part de la résolution nécessaire pour sonder des échelles spatiales dans la gamme précédemment mentionnée, les simulations du Full Universe Run sont à même d'explorer finement l'empreinte des non-linéarités sur ce signal. Les résultats et les détails d'une telle étude sont présentés dans l'article[79] sur les oscillations acoustiques des baryons au sein de la simulation DEUS-FUR ΛCDM fourni en annexe de ce manuscrit.

L'une des questions qui devient critique lorsqu'il s'agit de fournir des estimations précises à l'ordre du pourcent est celle de la façon de définir les pics des oscillations, ou dit autrement, celle de la façon de définir un spectre de référence non perturbé $P_{\mathrm{smooth}}(k)$. Différentes approches ont été employées dans la littérature en utilisant des modèles semi-analytiques[80], des interpolations à partir de données observationnelles[81, 82], ou en faisant évolué un spectre initial à l'aide d'une approche perturbative[83]. En définissant le spectre des oscillations par :

$$P_{\mathrm{BAO}}(k) = P(k) - P^{\mathrm{FUR}}_{\mathrm{smooth}}(k) \tag{2.23}$$

on extrait le signal de la figure 20. Quatre oscillations peuvent être clairement distinguées alors que dans le cas d'une simulation de 648 Mpc/h de côté, fourni ici à titre de référence, le même spectre apparait comme extrêmement bruité. Cette qualité de signal permet d'effectuer des comparaisons[79] avec les multiples prescriptions semi-analytiques existantes[84, 85, 86] destinées à modéliser les effets des non-linéarités sur les BAO. Aussi, l'impact de ces dernières peut être séparé en trois composantes :

$$P_{\mathrm{BAO}}^{\mathrm{fit}}(k) = \left[P^{\mathrm{lin}}\left(\frac{k}{\alpha(k)}\right) - P_{\mathrm{smooth}}^{\mathrm{lin}}\left(\frac{k}{\alpha(k)}\right)\right] e^{-\frac{\gamma^2(k)k^2}{2}} + \beta(k) \tag{2.24}$$

où :

- $P_{\mathrm{BAO}}^{\mathrm{fit}}$ est la modélisation du spectre numérique obtenu en figure 20
- P^{lin} est le spectre linéaire
- α modélise le « shift » horizontal des BAO
- β modélise le rôle d'un « continuum »
- γ modélise le « damping », l'amortissement des pics

Dans la pratique, à un redshift donné, on prend des polynômes du second ou troisième ordre pour α, β et γ. Cette modélisation en termes de shift, de continuum et de damping, bien qu'empirique et bien que les termes soient interdépendants, présente l'avantage de découpler les différents effets des non-linéarités. À titre indicatif, une erreur de 1% sur la position des pics à $z = 1$, c'est-à-dire sur le shift, peut entrainer une erreur de l'ordre de 4% sur l'estimation de l'équation d'état de l'énergie noire[78]. Cette modélisation ouvre la porte d'une part à des analyses plus poussées sur les biais induits sur les paramètres cosmologiques par la dynamique non-linéaire et d'autre part à l'étude de la dépendance cosmologique fine des déformations du profil des BAO en vue de s'en servir comme d'une nouvelle sonde de la structuration.

2.3.4 Propriétés des halos, l'exemple de la fonction de masse

En plus du spectre de puissance, les propriétés générales de la structuration peuvent être explorées à l'aide de la fonction de masse décrite par l'équation 2.21. Les fonctions de masse des trois modèles du Full Universe Run sont dépouillées en figure 22 et sont comparées à différents redshifts en figure 21. Dans les deux cas, celle-ci n'est mesurée que pour les masses supérieures à $10^{14}\,\mathrm{M}_\odot/\mathrm{h}$ du fait de la résolution en masse des simulations : cette limite correspond à des halos constitués d'une centaine de particules. À mesure que le redshift augmente, des halos de plus en plus nombreux sont formés et ceux-ci sont de plus en plus massifs. Explorer la forme de la fonction de masse pour des halos de plus de $10^{16}\,\mathrm{M}_\odot/\mathrm{h}$ demande de disposer d'un très grand volume étant donné la rareté de ces évènements.

Sur la figure 22, les premiers halos de plus de $10^{14}\,\mathrm{M}_\odot/\mathrm{h}$ sont détectés à $z \approx 3$. À $z = 0$, on en compte 145 millions pour ΛCDM, 169 millions pour WCDM mais seulement 76 millions pour RPCDM. Le halo le plus massif en cosmologie ΛCDM est d'environ $1.10 \times 10^{16}\,\mathrm{M}_\odot/\mathrm{h}$, tandis qu'il est de $1.23 \times 10^{16}\,\mathrm{M}_\odot/\mathrm{h}$ pour WCDM et $0.65 \times 10^{16}\,\mathrm{M}_\odot/\mathrm{h}$ pour RPCDM. L'analyse détaillée de l'histoire de formation de ces halos très massifs demanderait de construire des arbres de fusion comme en figure 16 afin de remonter à leurs progéniteurs. La comparaison en figure 21 vient confirmer ce qui avait été déduit des spectres de puissance à savoir que le modèle RPCDM forme moins de structures que le modèle de concordance tandis que WCDM est plus structuré que ce dernier. Les fonctions de multiplicité extraites de ces fonctions de masse permettraient d'étudier l'influence de l'énergie noire sur les processus d'effondrement. Dans la suite, disposer d'un modèle moins structuré que ΛCDM et d'un modèle plus structuré va permettre d'explorer plus efficacement l'effet de la structuration sur la métrique et donc la dépendance cosmologique de la propagation de la lumière à grande échelle.

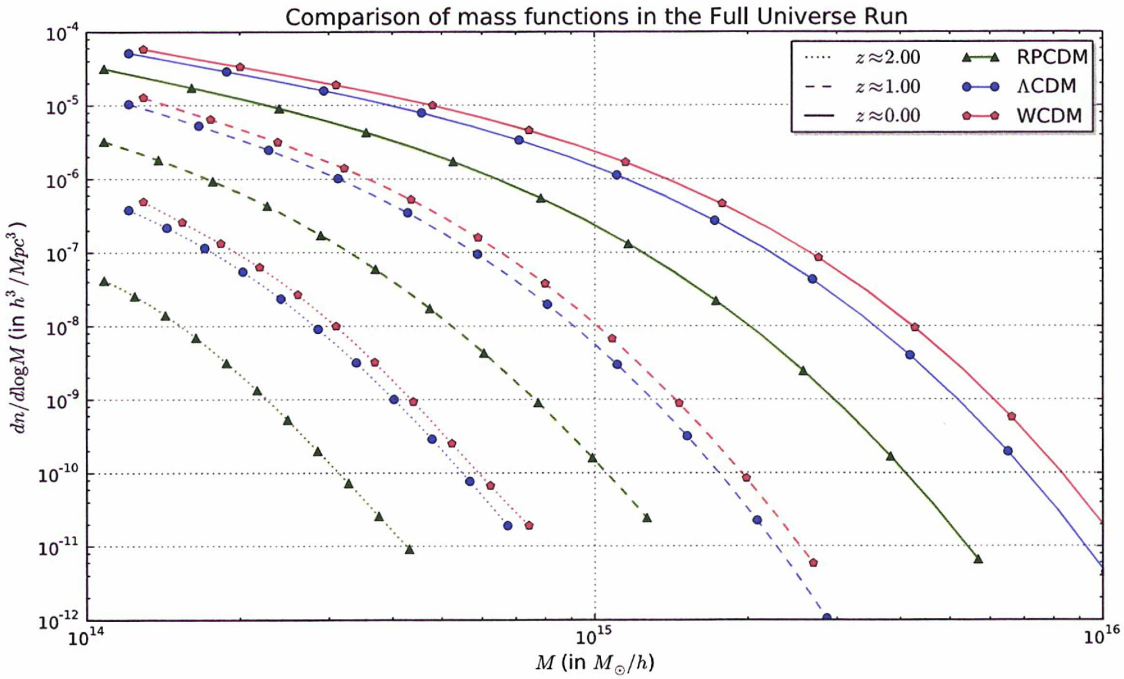


FIGURE 21 – Comparaison des fonctions de masse à $z = 2$, $z = 1$ et $z = 0$. Les fonctions de masse de la figure 22 sont comparées pour les trois modèles du Full Universe Run à trois redshifts différents. À mesure que le redshift évolue, chaque modèle forme de plus en plus de structures de plus en plus massives. Le modèle RPCDM est bien moins structuré que le modèle de concordance ΛCDM, lui-même moins structuré que WCDM. Ces différences importantes trouvent leur origine dans des histoires du facteur d'échelle légèrement différentes, mais qui marquent significativement le modèle hiérarchique de formation des structures de leur empreinte.

2.3.5 De la nécessité de nouvelles sondes dans l'espace des redshifts

Toutes les analyses décrites jusqu'à maintenant ont été effectuées dans l'espace comobile. Le spectre de puissance a pu être calculé pour différents temps cosmiques, de même que la fonction de masse et les oscillations acoustiques des baryons. Mais cela ne correspond pas à l'espace observable de la figure 13 : l'espace des redshifts, dans lequel l'Univers proche apparait comme très structuré et l'Univers lointain est quasi-homogène. Pour sonder les propriétés de l'accélération de l'expansion et contraindre les modèles théoriques associés, disposer de sondes dans le cône de lumière de l'observateur que nous sommes constituerait un atout conséquent en venant compléter l'approche, plus classique, basée sur l'étude de la structuration en coordonnées comobiles.

Au niveau observationnel, des propositions ambitieuses de mesure de redshift drift ont par exemple été formulées[87]. Il s'agirait, à l'ère des télescopes de très grande taille, de sonder directement la vitesse expansion par la mesure de :

$$\frac{\mathrm{d}z}{\mathrm{d}t_0} = (1+z)\,H_0 - H\,(z) \tag{2.25}$$

Dans le principe, cela consisterait à mesurer l'évolution du redshift de sources éloignées sur plusieurs années. Le très grand avantage de cette méthode est qu'elle permet de tester la cosmologie sans nécessiter d'hypothèse annexe, y compris si la cosmologie est inhomogène[88]. Mais pour parvenir à de tels résultats, cette méthode demande une très grande précision sur le redshift puisque qu'à titre d'exemple, la variation de ce dernier attendue sur 10 ans d'une source située à $z = 3$ est de l'ordre de la dizaine de cm/s. Une façon de parvenir à ce genre de précision serait d'observer un grand nombre de sources sur de longues durées, mais même avec de telles approches de nombreuses problématiques techniques, instrumentales et de traitement de données nécessiteraient encore d'être résolues.

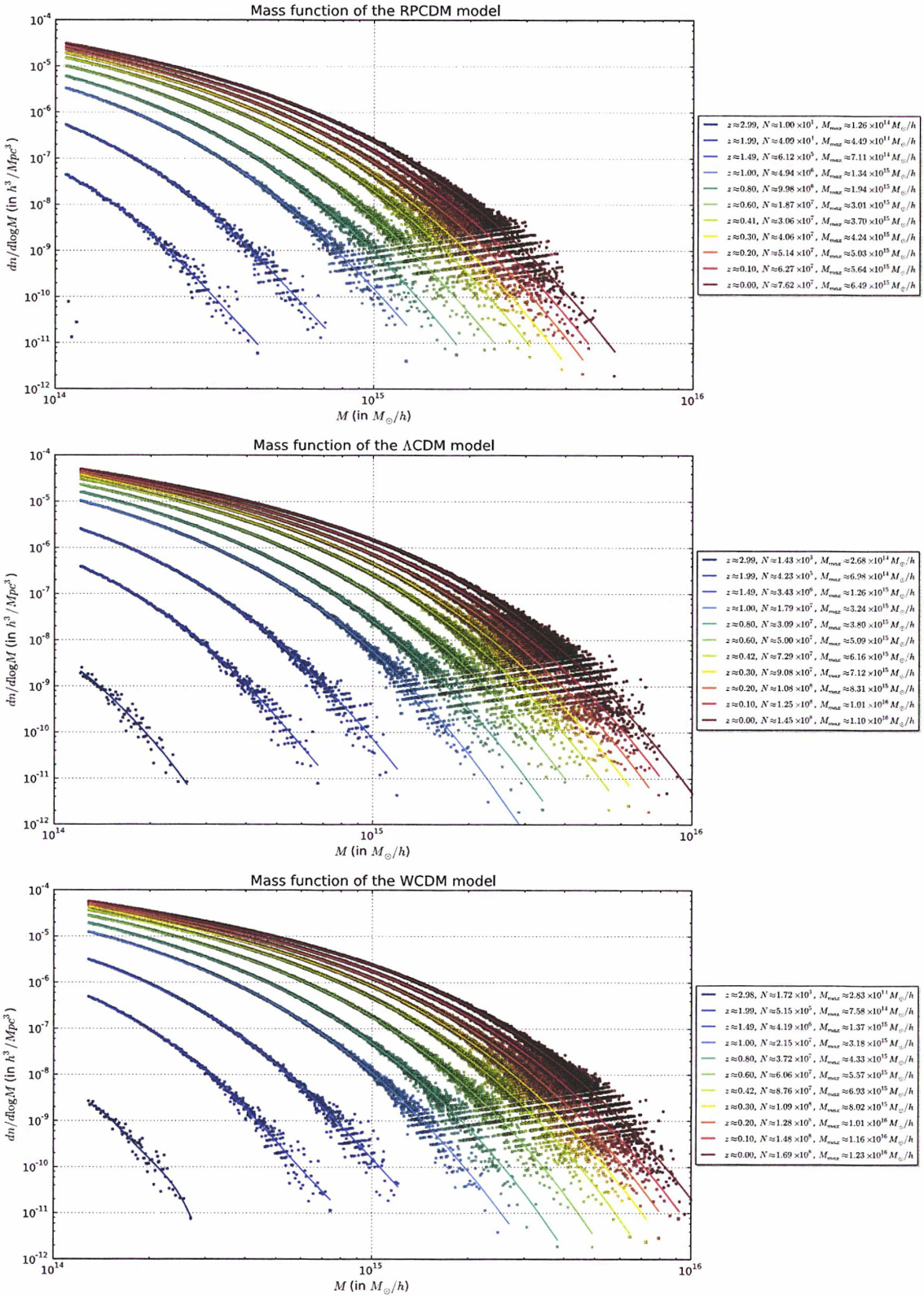


Figure 22 – Fonctions de masse numériques des simulations Full Universe Run. La fonction de masse est calculée conformément à l'équation 2.21 en comptant le nombre de halos d'une masse donnée dans chaque simulation. Chaque couleur est associée à un redshift et chaque marqueur correspond à un point de mesure de la fonction de masse. Les courbes en trait plein sont obtenues par des régressions polynomiales d'ordre 4. Dans la légende, N indique le nombre total de halos de plus de 100 particules et M_{max} correpond à la valeur du halo le plus massif à un redshift donné.

Sans aller jusqu'à la mesure de l'expansion « en temps réel », d'autres effets dans l'espace des redshifts pourraient permettre de tester la cosmologie. La propagation de la lumière en fait partie. En effet, du fait de la structuration, la lumière ne se propage pas de la même façon que dans un univers purement friedmannien : elle est déviée sous l'influence des potentiels locaux. Comme nous allons le voir dans la suite, cela peut avoir de multiples conséquences observationnelles, dont certaines sont bien observées, comme le lentillage gravitationnel par exemple. Mieux comprendre et modéliser la façon dont la lumière se propage dans un univers structuré est important pour mieux observer la structure à grande échelle, mais la propagation de la lumière est elle-même porteuse d'information. L'exploration de cette dernière piste demande d'une part de traiter analytiquement les perturbations de la métrique introduites par les potentiels gravitationnels locaux pour accéder aux équations qui régissent le trajet de la lumière au sein de la structure filamentaire de l'Univers et d'autre part de quantifier ces effets en n'analysant pas les simulations comme un observateur extérieur de boîtes comobiles comme cela a été le cas jusqu'à présent, mais plutôt comme un observateur virtuel intérieur mesurant les caractéristiques de la distribution de matière autour de lui. Et c'est justement à cela que seront consacrés les deux prochains chapitres.

Chapitre 3

L'Univers perturbé : de la métrique en champ faible aux équations géodésiques

***

3.1 Contexte et problématique : des inhomogénéités au lentillage gravitationnel 46
3.1.1 Inhomogénéités et cosmologie observationnelle 46
3.1.2 Approximations usuelles 51
3.1.3 Problématique 53
3.2 Perturbations cosmologiques engendrées par la structuration 54
3.2.1 Perturbation de la métrique et connexions affines 54
3.2.2 Tenseur d'Einstein et tenseur énergie-impulsion 56
3.2.3 Équations d'Einstein perturbées 58
3.2.4 Quantités calculées dans les simulations cosmologiques 61
3.3 Établissement des équations géodésiques sous-tendant la propagation de la lumière . . 61
3.3.1 Cas général 61
3.3.2 Cas des géodésiques nulles 64
3.3.3 Formulation alternative 64
3.3.4 Expression du redshift 65
3.4 Implications cosmologiques : formulations analytiques et interprétations 66
3.4.1 Correspondance entre distributions de matière simulée et observée 66
3.4.2 Effets sur le redshift et sur la distance de luminosité 66

Comme nous l'avons vu aux chapitres précédents, la relativité générale permet de décrire l'Univers comme un objet physique à part entière, ouvrant la voie à la cosmologie et à la caractérisation globale de son évolution. Ainsi, au premier ordre, on peut le considérer comme rigoureusement homogène et isotrope et envisager la description de sa dynamique à l'aune de la métrique de Friedmann-Lemaître-Robertson-Walker. Mais l'Univers n'est pas empli d'une poussière uniformément distribuée, et nous sommes même la preuve du contraire. En effet, au cours de son évolution, et à partir des fluctuations primordiales de densité, des objets se sont formés : étoiles, galaxies, amas, superamas, et filaments. Et ce sont eux que nous observons aujourd'hui et à partir desquels nous essayons de remonter à la métrique globale de l'Univers. Mais en toute rigueur, ce que nous dit $G_{\mu\nu} = T_{\mu\nu}$, c'est que le contenu matériel de l'Univers et la géométrie de l'espace-temps sont intrinsèquement liés. Si le premier n'est pas localement homogène du fait de la structuration décrite au chapitre précédent, la seconde n'a aucune raison de l'être non plus. Ainsi si l'on cherche à comprendre ce que nous disent réellement les observations, il faut aller au-delà des univers purement friedmanniens. Et une première étape dans cette voie est de

considérer de faibles perturbations à la métrique homogène. Dans ce chapitre, on se placera donc dans le contexte de la théorie des perturbations cosmologiques avec deux objectifs en tête : d'une part établir le lien entre le formalisme perturbatif et les quantités calculées dans les simulations cosmologiques en vue de l'exploitation de ces dernières et d'autre part obtenir un ensemble d'équations géodésiques permettant de suivre la propagation de la lumière dans un univers structuré. Établir le lien entre le potentiel gravitationnel et les surdensités Cela nous conduira à nous intéresser à l'équation de Poisson pour établir le lien entre le potentiel gravitationnel et les surdensités dans un Univers perturbé. Cela nous conduira également à un système d'équations rendant possible l'intégration des géodésiques nulles dans une distribution de potentiel arbitraire, tant que celle-ci est compatible avec l'hypothèse de faibles perturbations. Le but est donc de proposer une approche générique de raytracing en champ faible, extensible à des ordres supérieurs, en évitant un certain nombre d'approximations géométriques usuellement utilisées pour au contraire rester au plus près du cadre relativiste de la cosmologie.

Aussi, ce chapitre est divisé en quatre sections. La première dresse un rapide portrait des principaux effets des inhomogénéités sur les observations et la manière usuelle de les prendre en compte, ce qui permet au passage de poser la problématique du raytracing en champ faible. La seconde partie est consacrée à l'obtention des équations d'Einstein perturbées afin de donner une grille de lecture aux simulations cosmologiques dans ce formalisme. La troisième partie concerne la dérivation des équations géodésiques elles-mêmes dans différents systèmes de coordonnées. Enfin, la dernière section est consacrée à une discussion autour des implications cosmologiques de ces équations ainsi que leur mise en perspective au regard de travaux sur la même thématique.

3.1 Contexte et problématique : des inhomogénéités au lentillage gravitationnel

3.1.1 Inhomogénéités et cosmologie observationnelle

Introduction

Les grandes structures de l'Univers et les vides cosmiques, les surdensités et les sous-densités, perturbent la métrique. L'on pourrait se dire que, n'étant pas au voisinage d'objets compacts tels que des trous noirs, ces perturbations de nature relativistes doivent rester négligeables dans un contexte cosmologique. Sauf que, dans la pratique, elles ont des conséquences à grande échelle bien observables. Observables et observées : en effet, le nombre de programmes et de projets de cosmologie observationnelle basé sur l'analyse de ces effets n'a cessé de croître ces dernières années. À titre d'exemple, la mission spatiale Euclid sur laquelle se focalise toute l'attention d'une partie de la communauté consacrera l'utilisation du lentillage faible pour inférer les propriétés de la structuration jusqu'à $z = 2$.

Les effets des inhomogénéités sur les observables sont multiples et trouvent leur origine dans la modification de quatre caractéristiques physiques liées aux photons : la position angulaire d'arrivée, la distance parcourue dans l'espace, le temps de propagation et l'énergie, auxquels on pourrait adjoindre un cinquième élément à savoir la polarisation. Or seules les valeurs finales de ces quantités à la position de l'observateur sont mesurables. Dès lors, l'information cosmologique ne peut être extraite que d'effets différentiels entre photons ou de corrélations avec des observables indépendantes, comme les vitesses péculières de sources astrophysiques ou comme les valeurs des potentiels d'avant-plan. Et ce sont justement ces variations différentielles qui entraînent de nombreuses conséquences observationnelles : la fluctuation des mesures des distances cosmologiques, le lentillage gravitationnel, l'effet Sachs-Wolfe intégré, l'effet des champs de vitesse locaux sur le redshift, les « time delays »gravitationnels... Le premier, concerne simplement la différence qu'il existe entre un rayon lumineux qui se propage en ligne droite dans une métrique friedmannienne et un rayon qui subit les déviations entraînées par la succession de surdensités et de sous-densités sur la ligne de visée. Or le fait d'interpréter la propa-

gation perturbée comme une propagation dans un univers homogène entraîne des conséquences sur les mesures de distance. Le second, le lentillage gravitationnel, qu'il soit faible ou fort, modifie quant à lui le profil des faisceaux de lumière : ceux-ci peuvent subir des contraintes de magnification ou de cisaillement, créant, encore une fois, un biais sur l'estimation des distances. Enfin, l'effet Sachs-Wolfe intégré concerne l'impact de la variation des potentiels locaux du fait de l'accélération de l'expansion sur la lumière qui les traverse tandis que les champs de vitesses influent naturellement sur la mesure du redshift.

Dans la suite, on se place dans un contexte perturbatif : l'expansion de l'Univers est toujours décrite par une métrique homogène et isotrope d'arrière-plan, mais celle-ci se trouve être localement modifiée sous l'action d'un champ faible. Nous ne sommes donc pas dans un cadre de métrique réellement inhomogène comme peuvent l'être les modèles de Lemaître-Tolman-Bondi[89, 90], de Szekeres[91] ou les métriques Swiss-Cheese[92, 93]. Le même genre de traitement de la propagation de la lumière que celui qui est développé ici pourrait toutefois y être adapté en reprenant la même démarche.

Fluctuation des distances

Le premier effet qui vient à l'esprit lorsque l'on se place dans un contexte où la métrique n'est plus localement friedmannienne concerne la trajectoire des photons elle-même. Pour visualiser le problème de la déviation des rayons lumineux, il suffit de d'abord considérer une déflexion pour un objet isolé, pour une métrique de Schwarzschild par exemple, comme illustré en figure 23. Si un tel observateur interprète son observation dans une métrique de Minkowski, alors la position de l'objet qu'il en déduit est différente de sa position réelle. Dans le cadre cosmologique, il se produit exactement le même phénomène, sauf que cette fois la lumière se propage entre la source S et l'observateur O suivant les structures rencontrées, c'est-à-dire, dans un formalisme de champ faible, suivant la valeur du potentiel Φ local et de ses dérivées. En conséquence, l'intégrale spatiale curviligne le long de la trajectoire n'est plus la même que l'intégrale rectiligne entre S et O pour un univers homogène. De plus, la position d'un objet dans l'espace comobile (r, θ, φ) est traditionnellement déduite de son redshift et de sa position observée dans le ciel $\left(\tilde{z}, \tilde{\theta}, \tilde{\varphi}\right)$. Or, après déflexion, l'angle d'« impact » $\left(\tilde{\theta}, \tilde{\varphi}\right)$ n'est plus égal à l'angle sous lequel l'objet aurait été vu dans un univers rigoureusement homogène. En outre, comme le redshift dépend du rapport des métriques de la source et de l'observateur, il dépend des potentiels Φ_S et Φ_O et la relation avec le facteur d'échelle a ne tient plus :

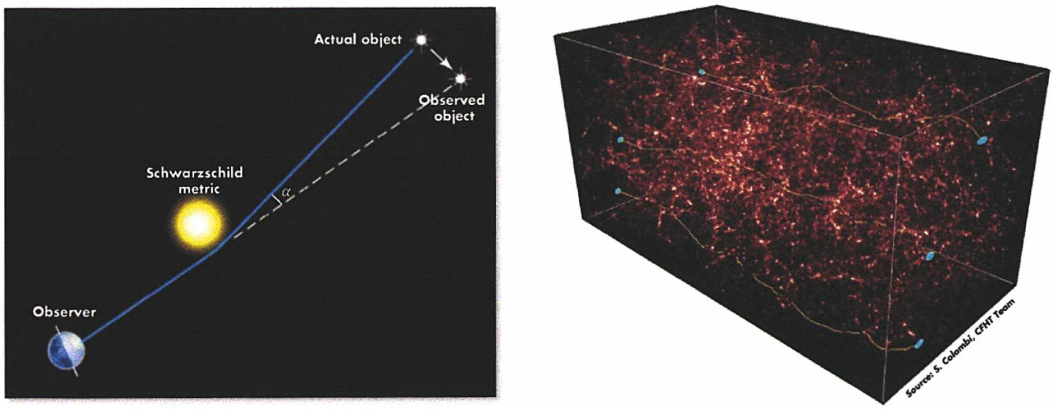
$$\boxed{1 + z = \frac{\left(g_{\mu\nu} k^\mu u^\nu\right)_S}{\left(g_{\mu\nu} k^\mu u^\nu\right)_O} \neq \frac{a_0}{a}} \tag{3.1}$$

En conséquence, la détermination des positions et des distances dans un univers structuré à partir des observations devient beaucoup plus subtil que dans le cadre d'un univers satisfaisant parfaitement le principe cosmologique.

Lentillage gravitationnel

Toutefois, la cosmologie observationnelle ne repose pas sur des trajectoires de photons prises isolément, mais plutôt sur des faisceaux de lumière provenant d'objets astrophysiques. Et de tels faisceaux vont justement être déformés par les structures, produisant des images magnifiées ou cisaillées des objets observés comme en figure 24. De plus, le calcul de la distance de luminosité, et donc aussi celui de la distance de diamètre angulaire, repose sur la « dilution » du flux lumineux le long de la trajectoire, soumis lui aussi aux inhomogénéités. Le formalisme du lentillage gravitationnel fournit le cadre nécessaire à l'étude de tous ces effets[94].

Le cas le plus simple de lentillage est celui d'une masse ponctuelle isolée comme celle de la figure 23, et un rayon lumineux de paramètre d'impact b avec b grand devant le rayon de Schwarzschild de



(a) Déflexion forte par un objet unique.

(b) Déflexion par une succession de structures.

FIGURE 23 – Illustration de la déflexion d'un rayon lumineux dans le cas d'une masse isolée et dans le cas cosmologique. À gauche, on considère le cas trivial de la déflexion d'un photon en métrique de Schwarzschild. Lorsque l'observateur réinterprète sa mesure en métrique de Minkowski, la position qu'il attribue à la source est différente de sa position réelle. Dans le cas cosmologique, à droite, le phénomène à l'œuvre est le même, sauf que cette fois le photon subit une succession de déflexions infinitésimales dues aux sous-densités et aux surdensités rencontrées localement. La réinterprétation des observations dans un cadre purement friedmannien peut alors introduire un biais sur la position des objets et leur distance.

l'objet défléchissant : $b \gg r_s = \frac{2GM}{c^2}$. L'angle de déflexion $\hat{\alpha}$ à l'infini prédit par la relativité générale vaut alors, en première approximation :

$$\boxed{\hat{\alpha} \approx \frac{4GM}{c^2 b}} \ll 1 \tag{3.2}$$

Cet angle dépend linéairement de M, si bien que si l'on considère une grille planaire d'objets défléchissants de masse M_i de coordonnées dans le plan $\vec{\xi}_i$, l'angle de déflexion d'un rayon de lumière croisant le plan à $\vec{\xi}$ s'écrit[95] :

$$\hat{\vec{\alpha}}\left(\vec{\xi}\right) = \sum_i \hat{\vec{\alpha}}_i\left(\vec{\xi} - \vec{\xi}_i\right) \approx \frac{4G}{c^2}\sum_i M_i \frac{\vec{\xi} - \vec{\xi}_i}{\left|\vec{\xi} - \vec{\xi}_i\right|^2} \tag{3.3}$$

Dans le cas d'une distribution tridimensionnelle de matière, comme un amas galactique, lorsque la source et l'observateur sont suffisamment éloignés de part et d'autre de l'objet, il est possible de considérer celui-ci comme plan : on parle alors d'approximation de lentille mince, illustrée en figure 25. Seule compte alors la distribution de matière projetée dans le plan :

$$\Sigma\left(\vec{\xi}\right) = \int \rho\left(\vec{\xi}, z\right) \mathrm{d}z \tag{3.4}$$

où z est la coordonnée perpendiculaire au plan. L'équation 3.3 peut alors se réécrire :

$$\boxed{\hat{\vec{\alpha}}\left(\vec{\xi}\right) \approx \frac{4G}{c^2}\int \frac{\vec{\xi} - \vec{\xi}'}{\left|\vec{\xi} - \vec{\xi}'\right|^2}\Sigma\left(\vec{\xi}'\right)\mathrm{d}^2\vec{\xi}'} \tag{3.5}$$

Dans ce cadre, et dans le cas où la déviation est petite, les angles $\vec{\theta}$, $\vec{\beta}$ et $\vec{\hat{\alpha}}$ repérés sur la figure 25 sont liés par :

$$\vec{\theta} D_s = \vec{\beta} D_s + \vec{\hat{\alpha}} D_{ds} \quad \Rightarrow \quad \vec{\beta} = \vec{\theta} - \vec{\alpha}\left(\vec{\theta}\right) \quad \text{avec} \quad \vec{\alpha}\left(\vec{\theta}\right) = \frac{D_{ds}}{D_s}\vec{\hat{\alpha}} \tag{3.6}$$

où $\vec{\alpha}$ est l'angle de déflexion réduit. La « multiple lens plane approximation » qui sera décrite ultérieurement dans ce chapitre s'appuie sur ces résultats pour proposer un traitement numérique de la

déviation de la lumière. La transformation de l'image de la source en l'image effectivement observée est donnée par la matrice de déformation :

$$\boxed{A_{ij} = \frac{\partial \beta_i}{\partial \theta_j}} \tag{3.7}$$

qui, pour une lentille unique et dans le cas où l'approximation de Born, qui consiste à considérer que les angles de déviation sont suffisamment petits pour que le potentiel gravitationnel responsable de la déflexion soit considéré le long de la ligne de visée et non pas le long de l'intégrale curviligne, est applicable peut s'écrire[96, 97] :

$$A = \begin{pmatrix} 1-\kappa & 0 \\ 0 & 1-\kappa \end{pmatrix} + \begin{pmatrix} -\gamma_1 & -\gamma_2 \\ -\gamma_2 & \gamma_1 \end{pmatrix} + \begin{pmatrix} 0 & \omega \\ -\omega & 0 \end{pmatrix} \quad \Rightarrow \quad \boxed{A = \begin{pmatrix} 1-\kappa-\gamma_1 & -\gamma_2+\omega \\ -\gamma_2-\omega & 1-\kappa+\gamma_1 \end{pmatrix}} \tag{3.8}$$

où κ est la convergence, $\gamma = \gamma_1 + i\gamma_2 = |\gamma|\, e^{i2\varphi}$ est le cisaillement, ou shear, et où ω est un terme de rotation du second ordre. À partir de ces quantités, et en négligeant ω, on peut définir la magnification μ et le shear réduit g :

$$\boxed{\mu = \frac{1}{\det A} = \frac{1}{(1-\kappa^2)-\gamma^2}} \quad \text{et} \quad \boxed{g = \frac{\gamma}{1-\kappa}} \tag{3.9}$$

Ces quantités sont au cœur du formalisme du lentillage gravitationnel car elles permettent de décrire et d'interpréter la façon dont un faisceau de lumière sera déformé par un objet massif le long du trajet de la lumière. Les déformations induites par la convergence sont isotropes, c'est-à-dire que la taille de l'image est multipliée par une constante dans toutes les directions, tandis que les déformations induites par le shear se font selon un axe particulier. Ainsi, un faisceau circulaire de rayon r devient, par lentillage, une ellipse de demi-grand axe a, de demi-petit axe b et d'ellipticité ϵ comme illustré en figure 25 :

$$a = \frac{r}{1-\kappa-\gamma} \qquad b = \frac{r}{1-\kappa+\gamma} \qquad \epsilon = \frac{a-b}{a+b} = \frac{\gamma}{1-\kappa} = g \tag{3.10}$$

La plupart des observations cosmologiques ayant trait au lentillage sont donc interprétées en ces termes. Il faut toutefois distinguer deux régimes distincts : le lentillage fort, qui impacte significativement des objets pris isolément, et le lentillage faible dont la détection repose sur des analyses statistiques. Dans la suite, et dans un contexte de nature cosmologique, on se place dans le second cas, c'est-à-dire pour :

$$\kappa, |\gamma| \ll 1 \tag{3.11}$$

Supposant une distribution de masse, il est possible de calculer les valeurs de la convergence et du shear, mais observationnellement, c'est la démarche inverse qui est menée : à partir de la mesure de la matrice de déformation, il est possible d'inférer les propriétés de la distribution de masse à grande échelle et donc de sonder la structuration sur des échelles cosmologiques. Toutefois, comme précisé en introduction, le lentillage ne fait pas que déformer les images des objets astrophysiques : il affecte également la mesure des distances de luminosité et des distances de diamètre angulaire en impactant le flux lumineux et l'angle sous lequel sont vus ces objets. Et interpréter des distances fondamentalement inhomogènes à l'aune de modèles d'Univers rigoureusement homogènes et isotropes peut naturellement impacter la mesure des paramètres cosmologiques.

Autres effets

En plus des deux effets décrits sur la mesure du redshift et la mesure des distances, les inhomogénéités ont trois autres conséquences observationnelles principales : l'effet des champs de vitesse locaux

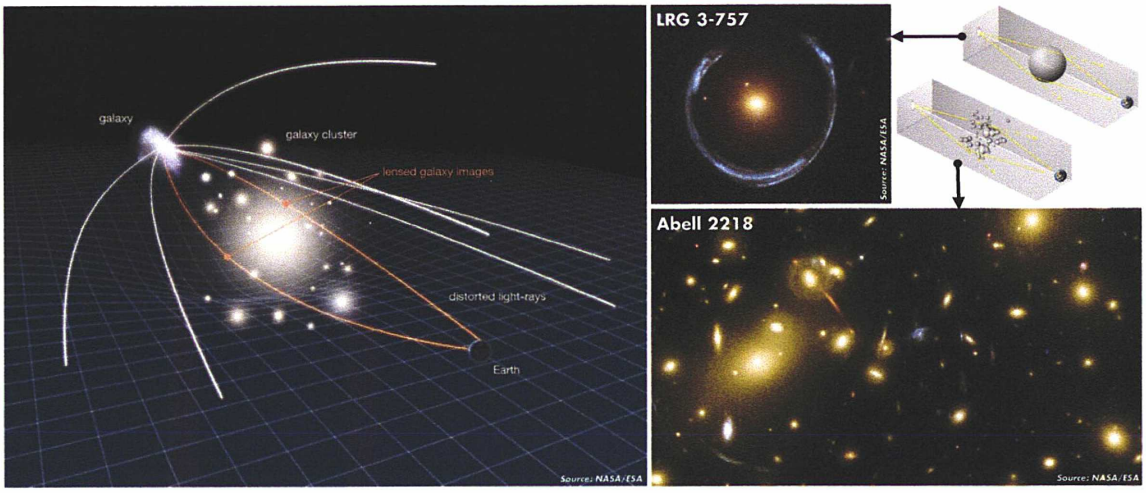
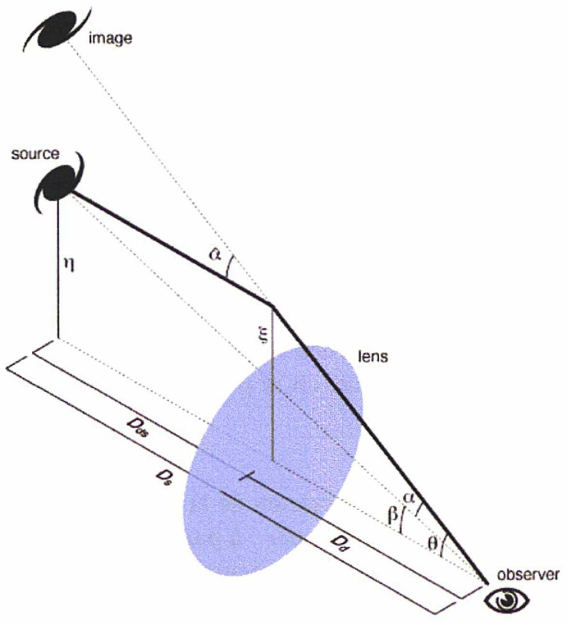
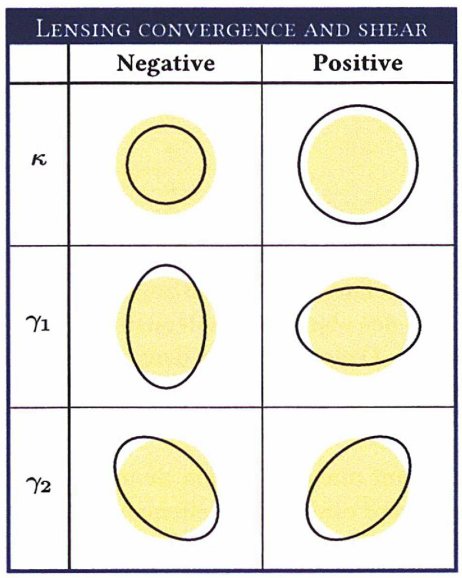


Figure 24 – Lentillage gravitationnel à l'échelle galactique et au-delà. À gauche est illustré le phénomène de lentillage d'une galaxie par un amas galactique d'avant-plan. En traversant la structure du cluster, le faisceau lumineux en provenance de la galaxie d'arrière-plan va être très largement déformé, produisant une image altérée de celle-ci pour l'observateur. Deux exemples de telles images sont présentés à droite. En haut le cas d'un anneau d'Einstein produit lorsque source, lentille et observateurs sont alignés et lorsque l'objet intermédiaire a une structure quasi-sphérique. Lorsque la structure de l'objet intermédiaire est plus complexe, comme dans le cas de l'amas Abell 2218, de multiples arcs peuvent être produits. Dans les deux cas, il s'agit de lentilles fortes. Le phénomène du lentillage faible repose sur le même principe, mais n'est pas détectable sur des sources isolées : sa détection repose sur des analyses statistiques pour mettre en évidence le même genre d'altérations sur un ensemble d'objets d'arrière-plan.



(a) Schéma de déflexion par une lentille mince (crédits : Michael Sachs).



(b) Effet de la convergence et du shear sur un objet circulaire.

Figure 25 – Traitement des déformations dans le formalisme du lentillage gravitationnel. Le schéma du modèle de lentille mince est fourni à gauche : la lumière d'une source S de paramètre d'impact η séparé de l'observateur par une distance D_s est déviée d'un angle $\hat{\alpha}$ par une lentille infiniment mince située à une distance D_d de l'observateur et D_{ds} de la source. La déformation d'un faisceau lumineux sous l'effet du lentillage, à droite, peut ensuite être décomposée en un terme de convergence κ, un terme de cisaillement (shear) de partie réelle γ_1 et de partie imaginaire γ_2 et un terme de rotation ω d'ordre plus élevé et non représenté ici. Ces termes permettent de spécifier la matrice de transformation A à appliquer aux coordonnées de la source pour obtenir l'image effectivement observée. L'impact des différents termes est ici illustré pour un faisceau originellement circulaire (en jaune).

sur le redshift, l'effet Sachs-Wolfe intégré et les « time delays » gravitationnels. Pour le premier, il s'agit juste de faire remarquer que les quadri-vitesses de la source et de l'observateur interviennent naturellement dans la formule du redshift 3.1. Et comme les sources sont toujours douées de vitesses péculières non-nulles par rapport à l'observateur dues à la dynamique locale, cela peut introduire un biais sur la mesure du facteur d'échelle à la position de l'objet si ces vitesses relatives ne sont pas préalablement soustraites des jeux de données observationnelles. Pour le second, il s'agit de la conséquence de la variation des potentiels locaux dans le temps sous l'effet de l'expansion et de son accélération[98]. En effet, en tombant dans le potentiel gravitationnel d'une grande structure, un photon gagne de l'énergie, qu'il perd quand il en ressort. Or, dans un univers en expansion accélérée, le potentiel est « dilué » entre le moment où le photon tombe dans le potentiel et le moment où il en ressort, entraînant au final un gain d'énergie. La situation est symétrique dans le cas de géodésiques nulles traversant des sous-densités. Cela entraîne, comme précédemment, une sous-évaluation ou une surévaluation du facteur d'échelle déduit directement du redshift mesuré. Quant aux troisièmes, les « time delays », aussi appelé effet Shapiro, sont liés au retard temporel accumulé par la lumière lorsque celle-ci est déviée au voisinage d'une masse. Aussi, dans le cadre d'un lentillage gravitationnel fort, les différentes images peuvent subir des retards différents. La mesure de ces derniers peut alors permettre d'apposer des contraintes sur la valeur de la constante de Hubble.

3.1.2 Approximations usuelles

Approche standard

Usuellement, tous les effets décrits précédemment son traités de façon indépendante. Observationnellement, bien entendu, mais aussi théoriquement et numériquement. Ce dernier aspect, l'aspect numérique, permet d'élargir le traitement théorique de ces phénomènes en cherchant, par exemple, à les quantifier au sein de simulations cosmologiques pour en extraire des effets plus fins que ce qu'une analyse purement analytique permettrait d'obtenir. Pour se faire, un certain nombre d'approximations, basées généralement sur le formalisme utilisé pour interpréter les observations, sont mises en œuvre.

Construction d'espaces de redshift

La première d'entre elle concerne l'obtention des données sur la structuration issues de simulations. En effet, suivre la propagation de la lumière d'une source à un observateur suppose d'avoir spécifier un observateur virtuel et construit son « cône de lumière ». Or le problème est que l'espace « naturel » des simulations cosmologiques est l'espace comobile, c'est-à-dire celui de tranches tridimensionnelles d'espace-temps à un temps cosmique donné. Comment passer de l'espace comobile à l'espace des redshifts d'un observateur virtuel ?

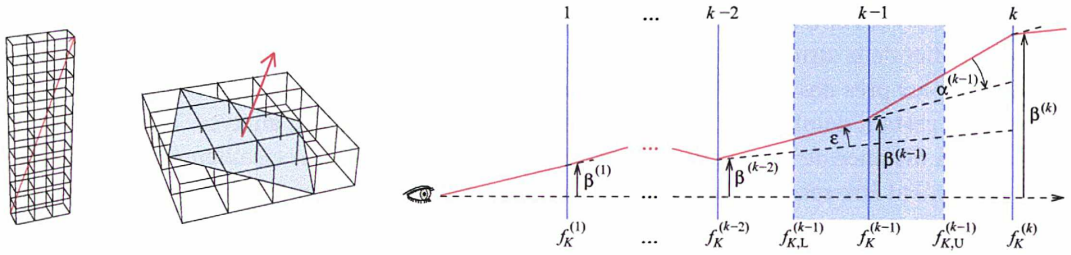
Plusieurs stratégies existent. La première, et sans doute la plus simple, consiste à ignorer le problème et à suivre des rayons lumineux dans l'espace comobile. Cette approximation peut se justifier dans le cadre d'une approche qui consiste à étudier la déviation des photons localement autour de l'observateur, jusqu'à un faible redshift. L'approche opposée consisterait à suivre directement la propagation de la lumière dans les codes dynamiques de formation des structures. C'est-à-dire, plus concrètement, lancer des particules de masse nulle dans toutes les directions à chaque fois que des structures se forment. Si cette approche est en principe celle qui permettrait le traitement le plus conforme à la réalité, elle nécessiterait une capacité de calcul hors de portée à l'heure actuelle : suivre des photons émis dans toutes les directions, pour toutes les sources et tout au long des simulations n'est pas viable en pratique, du moins actuellement. Cela pose d'ailleurs la question du sens de la propagation. En effet, dans cette dernière approche, les photons sont émis à partir des sources et voyagent librement au travers des simulations : on parle de « forward raytracing ». Mais dans ce cas il est impossible de prédire *a priori* la position de l'observateur que rencontrera chaque particule de lumière : il est donc nécessaire

de traiter tous les observateurs sur un pied d'égalité en laissant les particules se propager dans toute la simulation. Il est toutefois possible d'opter pour une approche de « backward raytracing » : dans ce cas on spécifie l'observateur *a priori* et l'on cherche à remonter aux sources en propageant les faisceaux lumineux à rebours. Cette approche demande de traiter beaucoup moins de photons que la précédente, ce qui rend possible son implémentation et son exécution sur les machines actuelles. Deux méthodes peuvent alors être définies, l'une étant plus coûteuse que l'autre en termes numériques et en termes d'espace de stockage. En effet, la spécification d'un observateur n'implique pas que l'on connaisse son cône de lumière avant d'avoir calculé les géodésiques nulles dans l'univers structuré qui est le sien. La façon la plus rigoureuse de déterminer le cône de lumière en question est donc de stocker l'intégralité de l'espace-temps 4-dimensionnel, c'est-à-dire l'intégralité de l'évolution de la structuration dans toute la simulation, et d'étudier les géodésiques nulles dans cet espace. Mais tout stocker peut vite devenir prohibitif. Aussi, en supposant que les déviations restent faibles et que les potentiels évoluent lentement, il est possible de pré-sélectionner les données dignes d'intérêt au cours de la simulation. Cela revient à construire la surface du cône de lumière en supposant que la structuration vue par des photons déviés n'a pas eu le temps d'évoluer par rapport à celle qui serait vue par des photons se propageant dans un univers homogène. En régime de lentillage gravitationnel faible, cette approximation semble largement justifiée et elle permet de garantir qu'un photon passe à la fois par une source et un observateur donnés.

Cette dernière solution est la plus utilisée. Cependant, même dans ce cadre, plusieurs techniques existent pour construire l'espace des redshifts à partir de cubes comobiles simulés à différents facteurs d'échelle. Lorsque l'on travaille avec des boîtes de simulation d'au plus 1 $\mathrm{Gpc/h}$ de côté, ce qui constitue la très grande majorité des cas, l'espace disponible est trop petit pour y loger l'entièreté d'un cône. La stratégie consiste alors à construire l'espace des redshifts à partir d'un pavage de boîtes comobiles à différents facteurs d'échelle, d'autant plus petits que l'on s'éloigne de l'observateur. C'est par exemple la stratégie suivie par Hilbert *et al.*[99] et Killedar *et al.*[100] pour effectuer un raytracing au travers de simulations cosmologiques, comme illustré en figure 26. Pour minimiser le risque qu'un faisceau ne rencontre plusieurs fois les mêmes structures à différents redshifts, les boîtes sont pivotées au sein du pavage et les rayons ne sont jamais lancés parallèlement aux arrêtes des cubes. Toutefois, de tels pavages ne permettent en général pas d'explorer de grands volumes dans toutes les directions. Ainsi, seule une petite partie du ciel de l'observateur virtuel est étudiée : par exemple une zone de $0.33\,^\circ \times 0.33\,^\circ$ jusqu'à $z = 0.5$ dans Killedar *et al.*[100], ce qui peut limiter la qualité des analyses statistiques effectuées dans ces espaces de redshift. De plus, le seul fait de répliquer une boîte de simulation à différents redshifts introduit nécessairement des corrélations artificielles du fait des couplages d'échelles qui existent, même si de tels effets tentent d'être contenus autant que faire se peut par l'orientation des rayons et du pavage. Enfin, une telle méthode de juxtaposition de boîtes à des temps cosmiques différents peut introduire une discrétisation du redshift avec un pas $\mathrm{d}z$ non négligeable. Au prix de ces quelques approximations, cette méthode de construction de cônes reste sans doute la plus à même de permettre l'exploration de la propagation de la lumière jusqu'à des redshifts élevés à l'aide de simulations cosmologiques de quelques dizaines ou centaines de $\mathrm{Mpc/h}$ de côté.

Termes de couplages et approximation de Born

Au niveau analytique, cette fois, deux approximations sont usuellement utilisées pour mener à bien les calculs, notamment dans le formalisme du lentillage gravitationnel. La première consiste à négliger les termes de couplage multi-lentilles : on considère comme indépendantes des lentilles situées à deux redshifts z_1 et z_2 différents et rencontrées successivement par un faisceau lumineux. Dit autrement, le fait que le faisceau soit déjà déformé par une structure à haut redshift lorsqu'il en rencontre une seconde à un redshift inférieur est négligé. Cette approximation tient lorsque les masses défléchissantes sont confinées à un domaine de redshift particulier, mais lorsqu'elles sont distribuées régulièrement le long de la ligne de visée les termes de couplages deviennent nécessairement plus importants, limitant du même coup la portée de l'approximation. D'autre part, l'approximation de Born, consiste à considérer que le potentiel gravitationnel vu par un photon défléchi est celui rencontré sur la trajectoire du même



(a) Construction d'un cône à partir d'un pavage de boîtes de simulations.

(b) Traitement de la déflexion par une succession de plans de lentillage.

Figure 26 – Méthodes de backward raytracing mise en œuvre par Hilbert *et al.*[99]. Pour construire le cône de lumière d'un observateur virtuel à l'aide de boîtes de simulation de 500 Mpc/h de côté, les auteurs de l'article réalisent un pavage comme présenté sur la figure de gauche. Cette méthode de réplication est souvent utilisée pour produire des espaces de redshifts à partir des données de simulation. Pour éviter qu'un rayon lumineux ne subisse plusieurs fois le potentiel de la même structure à différents redshifts, les cubes sont pivotés différemment au sein du pavage et les rayons sont lancés dans des directions non parallèles aux arrêtes des cubes. Chaque rayon lancé depuis l'observateur jusqu'aux sources est ensuite dévié par une succession de plans de lentillage, comme illustré à droite. La densité de la structuration est projetée sur chacun de ces plans pour calculer la valeur du potentiel gravitationnel intervenant dans la déviation du rayon.

photon non défléchi. Pour des lentilles uniques et en régime de lentillage faible, cela ne pose pas de problème. Mais, là encore, lorsqu'il s'agit de considérer le cas de multiples lentilles successives sur de grandes échelles, les différences entre le potentiel effectivement rencontré sur la trajectoire et le potentiel rencontré sur la ligne de visée rectiligne peuvent devenir significatives.

Multiple lens plane approximation

Pour éviter de négliger les deux effets décrits dans le paragraphe précédent, une stratégie souvent mise en place dans les simulations de lentillage faible est la « multiple lens plane approximation », mise en œuvre par exemple dans l'approche de Hilbert *et al.*[99]. Au lieu de considérer un potentiel continu le long de la trajectoire des rayons lumineux, on considère que les déviations sont le résultat d'une succession de lentilles minces. La densité de matière est alors projetée sur les plans de ces lentilles minces pour y calculer le potentiel défléchissant comme illustré en figure 26. Dans l'article de Hilbert[99], une trentaine de plans sont utilisés pour des sources situées à $z = 3$. Si elle permet d'aller plus loin que l'approximation de Born, et si elle permet de prendre en compte les termes de couplages multilentilles, la « Multiple lens plane approximation » reste toutefois inscrite dans le cadre de l'approximation des lentilles minces. De plus, la matrice de distorsion est généralement calculée au premier ordre, en assimilant en outre la dérivée perpendiculaire au chemin du photon à la dérivée perpendiculaire à la direction initiale de propagation. L'approche de Killedar *et al.*[100] tente de dépasser ces approximations par le calcul du potentiel dans un maillage cubique à partir de la distribution de matière issue des simulations. Ce maillage reste toutefois régulier et ne permet pas de sonder les zones plus structurées avec plus de précision au moyen de raffinement adaptatif.

3.1.3 Problématique

La propagation de la lumière dans un univers structuré peut donc être liée à de nombreux effets observationnels parmi lesquels on trouve le lentillage, l'introduction d'un biais dans la mesure des distances cosmologiques et l'effet Sachs-Wolfe intégré. L'analyse de chacun de ces effets repose usuellement sur des approches et des formalismes différents. En outre, dans le cas du traitement du lentillage faible au niveau analytique et au niveau numérique, un certain nombre d'approximations de nature géométrique sont mises en œuvre, à l'image de celles qui sont utilisées pour traiter et interpréter les observations de lentillage.

Pourtant, tous ces effets ne sont que les multiples conséquences des déformations des géodésiques nulles sous l'effet de la structuration. Ne serait-il donc pas possible d'en proposer un traitement unifié en restant au plus près des équations de la relativité générale ? Cela permettrait, du même coup, de se détacher d'un certain nombre d'approximations propres à chaque type d'analyse. Tous les effets sur le redshift, sur la distance de diamètre angulaire, sur la magnification et le cisaillement des faisceaux de lumière pourraient être retrouvés naturellement *a posteriori* de ce traitement unifié et non pas supposés *a priori*.

Dans cet objectif, ce chapitre se focalise sur les aspects analytiques autour de la théorie des perturbations cosmologiques. Le but est de parvenir à un système d'équations géodésiques permettant un traitement unifié de la propagation de la lumière pour une distribution de potentiel arbitraire, que celle-ci provienne directement de simulations numériques ou que celle-ci soit reconstruite à partir d'observations. Le chapitre suivant sera consacré à l'étude des simulations du Full Universe Run à l'aide du formalisme développé ici. Cette approche est résumée de manière graphique en figure 27.

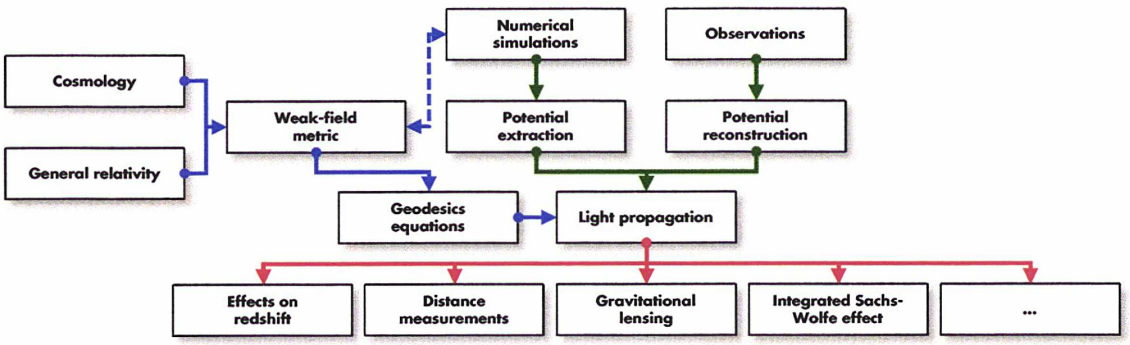


Figure 27 – Approche unifiée du traitement de la propagation de la lumière. Au lieu de chercher à étudier un effet particulier et d'appliquer les approximations adéquates *a priori*, on cherche à traiter la problématique de la propagation de la lumière dans un univers structuré de la façon la plus générale possible. Cela passe notamment par l'obtention des équations géodésiques en métrique faiblement perturbée et par l'établissement des liens avec les quantités calculées dans les simulations cosmologiques. Cette approche générique applicable à une distribution arbitraire de potentiel permet *a posteriori* d'aborder toutes les problématiques liées au trajet de la lumière, de la mesure des distances cosmologiques jusqu'à l'effet Sachs-Wolfe intégré. La démarche est de plus généralisable à toute métrique et au cas $\mathrm{d}s^2 \neq 0$.

3.2 Perturbations cosmologiques engendrées par la structuration

3.2.1 Perturbation de la métrique et connexions affines

Dans la suite, on considère une perturbation $\delta g_{\mu\nu}$ à un fond de métrique homogène $\bar{g}_{\mu\nu}$ modélisant au premier ordre les fluctuations locales de la métrique entrainées par la formation des grandes structures. Ce terme dépend du temps, comme la métrique homogène d'un univers en expansion, mais aussi des coordonnées d'espace. Le tenseur global considéré peut donc s'écrire :

$$\boxed{g_{\mu\nu}(t,x) = \bar{g}_{\mu\nu}(t) + \delta g_{\mu\nu}(t,x_i)} \tag{3.12}$$

La métrique homogène considérée ici est une métrique de Friedmann-Lemaître-Robertson-Walker que l'on suppose, de courbure nulle afin de traiter le problème en coordonnées cartésiennes, comme cela est le cas pour les simulations cosmologiques :

$$\mathrm{d}s^2 = -c^2\,\mathrm{d}t^2 + a(t)^2\,\mathrm{d}\Sigma^2 \qquad \text{avec} \qquad \mathrm{d}\Sigma^2 = \mathrm{d}x^2 + \mathrm{d}y^2 + \mathrm{d}z^2 \tag{3.13}$$

Les perturbations, elles, peuvent être de trois natures différentes : scalaires, vectorielles ou tensorielles. Ces trois types de perturbations du fond FLRW peuvent être considérées comme indépendants, ce

qui permet d'effectuer une décomposition scalaire-vecteur-tenseur. Dans toute la suite, on considère seulement des perturbations scalaires pouvant être associées aux sous et surdensités apparaissant lors du processus de formation des structures. Il ne s'agit en effet ni de traiter le cas des perturbations vectorielles, ces dernières s'atténuant avec le temps dans un Univers en expansion, ni de traiter le cas des perturbations tensorielles et des ondes gravitationnelles correspondantes. Dans le cas le plus général, la métrique perturbée peut s'écrire[3] :

$$\mathrm{d}s^2 = -c^2\left(1+2\frac{\Psi}{c^2}\right)\mathrm{d}t + 2\frac{c}{a}B_{,i}\,\mathrm{d}x^i\,\mathrm{d}t + a^2\left[\left(1-2\frac{\Phi}{c^2}\right)\delta_{ij} + 2E_{,ij}\right]\mathrm{d}x^i\,\mathrm{d}x^j \tag{3.14}$$

où Ψ est appelé le lapse, Φ la perturbation de la courbure spatiale, B le shift et E le shear, tous les quatre étant dépendants à la fois des coordonnées de temps et d'espace. Certaines de ces quantités dépendent de la jauge dans laquelle on se place. On peut montrer que par un choix judicieux de jauge il est possible d'annuler certains des termes perturbatifs. En particulier, dans la jauge dite newtonienne, les termes de shift et de shear s'annulent si bien qu'il est possible d'écrire la métrique sous la forme :

$$\boxed{\begin{aligned}\mathrm{d}s^2 &= -c^2\left(1+2\frac{\Psi}{c^2}\right)\mathrm{d}t^2 + a^2\left(1-2\frac{\Phi}{c^2}\right)\left(\mathrm{d}x^2+\mathrm{d}y^2+\mathrm{d}z^2\right)\\ &= a^2\left[-c^2\left(1+2\frac{\Psi}{c^2}\right)\mathrm{d}\eta^2 + \left(1-2\frac{\Phi}{c^2}\right)\left(\mathrm{d}x^2+\mathrm{d}y^2+\mathrm{d}z^2\right)\right]\end{aligned}} \tag{3.15}$$

De la même manière que la métrique $g_{\mu\nu}$ est obtenue comme la somme de la métrique homogène $\bar{g}_{\mu\nu}$ et d'un terme perturbatif $h_{\mu\nu} = \delta g_{\mu\nu}$, la métrique inverse $g^{\mu\nu}$ peut s'écrire :

$$g^{\mu\nu} = \bar{g}^{\mu\nu} + k^{\mu\nu} \tag{3.16}$$

où $k^{\mu\nu}$ est un terme perturbatif. On a alors :

$$g_{\alpha\beta}g^{\beta\gamma} = \delta^\gamma_\alpha + h_{\alpha\beta}\bar{g}^{\beta\gamma} + \bar{g}_{\alpha\beta}k^{\beta\gamma} + h_{\alpha\beta}k^{\beta\gamma} \tag{3.17}$$

si bien que l'annulation des termes du premier ordre donne :

$$h_{\alpha\beta}\bar{g}^{\beta\gamma} + \bar{g}_{\alpha\beta}k^{\beta\gamma} = 0 \quad\Rightarrow\quad k^{\beta\gamma} = -\bar{g}^{\delta\alpha}h_{\alpha\beta}\bar{g}^{\beta\gamma} \quad\Rightarrow\quad \begin{cases} k^{00} = 2\dfrac{\Psi}{c^4} \\ k^{ii} = 2\dfrac{\Phi}{a^2c^2}\end{cases} \tag{3.18}$$

soit les expressions pour $g_{\mu\nu}$ en temps cosmique et en temps conforme présentées en figure 28. Connaissant la métrique et la métrique inverse, il est alors possible de calculer les connexions affines :

$$\boxed{\Gamma^\alpha_{\beta\gamma} = \frac{1}{2}g^{\alpha\delta}\left(\frac{\partial g_{\delta\beta}}{\partial x^\gamma} + \frac{\partial g_{\delta\gamma}}{\partial x^\beta} - \frac{\partial g_{\beta\gamma}}{\partial x^\delta}\right)} \underset{\text{diag}}{=} \begin{cases} 0 \text{ si } \alpha\neq\beta\neq\gamma \\ \frac{1}{2}g^{\alpha\alpha}\left(\frac{\partial g_{\alpha\beta}}{\partial x^\gamma} + \frac{\partial g_{\alpha\gamma}}{\partial x^\beta} - \frac{\partial g_{\beta\gamma}}{\partial x^\alpha}\right) \text{ sinon}\end{cases} \tag{3.19}$$

où les expressions présentées à droite correspondent au cas d'un tenseur métrique diagonal, comme c'est le cas ici. Les développements pour la métrique 3.15 conduisent alors aux résultats suivants :

$$\begin{aligned}
\Gamma^0_{00} &= \frac{1}{c^2}\frac{\partial\Psi}{\partial t} - \frac{2}{c^4}\Psi\frac{\partial\Psi}{\partial t} \\
\Gamma^i_{ii} &= -\frac{1}{c^2}\frac{\partial\Phi}{\partial x^i} - \frac{2}{c^4}\Phi\frac{\partial\Phi}{\partial x^i} \\
\Gamma^0_{ii} &= \frac{a\dot{a}}{c^2} - \frac{2a\dot{a}}{c^4}(\Psi+\Phi) - \frac{a^2}{c^4}\frac{\partial\Phi}{\partial t} + \frac{4a\dot{a}}{c^6}\Psi\Phi + \frac{2a^2}{c^6}\Psi\frac{\partial\Phi}{\partial t} \\
\Gamma^i_{00} &= \frac{1}{a^2}\frac{\partial\Psi}{\partial x^i} + \frac{2}{a^2c^2}\Phi\frac{\partial\Psi}{\partial x^i} \\
\Gamma^0_{0i} = \Gamma^0_{i0} &= \frac{1}{c^2}\frac{\partial\Psi}{\partial x^i} - \frac{2}{c^4}\Psi\frac{\partial\Psi}{\partial x^i} \\
\Gamma^i_{i0} = \Gamma^i_{0i} &= \frac{\dot{a}}{a} - \frac{1}{c^2}\frac{\partial\Phi}{\partial t} - \frac{4\dot{a}}{ac^4}\Phi^2 - \frac{2}{c^4}\Phi\frac{\partial\Phi}{\partial t} \\
\Gamma^i_{jj} &= \frac{1}{c^2}\frac{\partial\Phi}{\partial x^i} + \frac{2}{c^4}\Phi\frac{\partial\Phi}{\partial x^i} \\
\Gamma^i_{ij} = \Gamma^i_{ji} &= -\frac{1}{c^2}\frac{\partial\Phi}{\partial x^j} - \frac{2}{c^4}\Phi\frac{\partial\Phi}{\partial x^j}
\end{aligned} \tag{3.20}$$

Dans toute la suite, on néglige les termes de second ordre en Ψ, Φ et leurs dérivées respectives : dans ce cas les expressions des symboles de Christoffel se ramènent à celles présentées en figure 3.15. Un signe « $=$ » signifiera donc une égalité des termes de premier ordre entre deux expressions. Toutefois, cela implique que des termes en $\Phi\nabla^2\Phi$ seront également négligés par rapport à Φ. Cette approximation est consistante avec les simulations cosmologiques, et permet de se ramener à des expressions indépendantes de jauge. Cependant, elle nécessiterait d'être testée[101] en reprenant la même démarche mais en gardant, cette fois, les termes du second ordre. Cela pourra faire l'objet de développements ultérieurs.

3.2.2 Tenseur d'Einstein et tenseur énergie-impulsion

On cherche maintenant à déterminer les composantes du tenseur d'Einstein et du tenseur énergie-impulsion. Le premier se déduit directement de la métrique et des connexions affines, en calculant d'abord le tenseur de Ricci et le scalaire de courbure :

$$R_{\mu\nu} = \frac{\partial\Gamma^\rho_{\nu\mu}}{\partial\rho} - \frac{\partial\Gamma^\rho_{\rho\mu}}{\partial\nu} + \Gamma^\rho_{\rho\lambda}\Gamma^\lambda_{\nu\mu} - \Gamma^\rho_{\nu\lambda}\Gamma^\lambda_{\rho\mu} \qquad \text{et} \qquad R = g^{\mu\nu}R_{\mu\nu} \tag{3.21}$$

puis en déterminant :

$$\boxed{G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2}g_{\mu\nu}R} \tag{3.22}$$

Le détail des composantes ainsi obtenues pour le premier ordre est donné en figure 29.

De la même manière que l'on a considéré une métrique friedmannienne faiblement perturbée, on s'intéresse aux perturbations affectant le tenseur énergie-impulsion $T_{\mu\nu}$. Dans ce contexte, on considère un fluide parfait auquel on applique un traitement perturbatif à toutes les quantités, c'est-à-dire :

$$\begin{aligned}
T_{\mu\nu} &= \left(\rho c^2 + P\right)u_\mu u_\nu + P g_{\mu\nu} \\
T^{\mu\nu} &= \left(\rho c^2 + P\right)u^\mu u^\nu + P g^{\mu\nu} \\
T^\mu_\nu &= \left(\rho c^2 + P\right)u^\mu u_\nu + P\delta^\mu_\nu
\end{aligned} \quad \text{avec} \quad \begin{aligned}
\rho &= \bar{\rho} + \delta\rho \\
P &= \bar{P} + \delta P \\
u_\mu &= \bar{u}_\mu + \delta u_\mu \\
g_{\mu\nu} &= \bar{g}_{\mu\nu} + \delta g_{\mu\nu}
\end{aligned} \quad \text{et} \quad \boxed{T_{\mu\nu} = \bar{T}_{\mu\nu} + \delta T_{\mu\nu}} \tag{3.23}$$

où les quantités $\bar{X}$ sont les quantités moyennes uniquement dépendantes du temps et où les δX représentent les perturbations locales dépendantes du temps t et des coordonnées d'espace x, y et z. Dans

Weak-field metric		
	Cosmic time	**Conformal time**
g_{00}	$-c^2\left(1+2\frac{\Psi}{c^2}\right)$	$-a^2c^2\left(1+2\frac{\Psi}{c^2}\right)$
g_{ii}	$a^2\left(1-2\frac{\Phi}{c^2}\right)$	$a^2\left(1-2\frac{\Phi}{c^2}\right)$
g^{00}	$-\frac{1}{c^2}\left(1-2\frac{\Psi}{c^2}\right)$	$-\frac{1}{a^2c^2}\left(1-2\frac{\Psi}{c^2}\right)$
g^{ii}	$\frac{1}{a^2}\left(1+2\frac{\Phi}{c^2}\right)$	$\frac{1}{a^2}\left(1+2\frac{\Phi}{c^2}\right)$

Derivatives of the weak-field metric		
	Cosmic time	**Conformal time**
$\frac{\partial g_{00}}{\partial x^0}$	$-2\frac{\partial\Psi}{\partial t}$	$-2aa'c^2-4aa'\Psi-2a^2\frac{\partial\Psi}{\partial\eta}$
$\frac{\partial g_{00}}{\partial x^i}$	$-2\frac{\partial\Psi}{\partial x^i}$	$-2a^2\frac{\partial\Psi}{\partial x^i}$
$\frac{\partial g_{ii}}{\partial x^0}$	$2a\dot{a}-\frac{4a\dot{a}}{c^2}\Phi-\frac{2a^2}{c^2}\frac{\partial\Phi}{\partial t}$	$2aa'-\frac{4aa'}{c^2}\Phi-\frac{2a^2}{c^2}\frac{\partial\Phi}{\partial\eta}$
$\frac{\partial g_{ii}}{\partial x^i}$	$-\frac{2a^2}{c^2}\frac{\partial\Phi}{\partial x^i}$	$-\frac{2a^2}{c^2}\frac{\partial\Phi}{\partial x^i}$
$\frac{\partial g_{ii}}{\partial x^j}$	$-\frac{2a^2}{c^2}\frac{\partial\Phi}{\partial x^j}$	$-\frac{2a^2}{c^2}\frac{\partial\Phi}{\partial x^j}$

Affine connections at first order		
	Cosmic time	**Conformal time**
Γ^0_{00}	$\frac{1}{c^2}\frac{\partial\Psi}{\partial t}$	$\frac{a'}{a}+\frac{1}{c^2}\frac{\partial\Psi}{\partial\eta}$
Γ^i_{ii}	$-\frac{1}{c^2}\frac{\partial\Phi}{\partial x^i}$	$-\frac{1}{c^2}\frac{\partial\Phi}{\partial x^i}$
Γ^0_{ii}	$\frac{a\dot{a}}{c^2}-\frac{2a\dot{a}}{c^4}(\Psi+\Phi)-\frac{a^2}{c^4}\frac{\partial\Phi}{\partial t}$	$\frac{a'}{ac^2}-\frac{2a'}{ac^4}(\Psi+\Phi)-\frac{1}{c^4}\frac{\partial\Phi}{\partial\eta}$
Γ^i_{00}	$\frac{1}{a^2}\frac{\partial\Psi}{\partial x^i}$	$\frac{\partial\Psi}{\partial x^i}$
$\Gamma^0_{0i}=\Gamma^0_{i0}$	$\frac{1}{c^2}\frac{\partial\Psi}{\partial x^i}$	$\frac{1}{c^2}\frac{\partial\Psi}{\partial x^i}$
$\Gamma^i_{i0}=\Gamma^i_{0i}$	$\frac{\dot{a}}{a}-\frac{1}{c^2}\frac{\partial\Phi}{\partial t}$	$\frac{a'}{a}-\frac{1}{c^2}\frac{\partial\Psi}{\partial\eta}$
Γ^i_{jj}	$\frac{1}{c^2}\frac{\partial\Phi}{\partial x^i}$	$\frac{1}{c^2}\frac{\partial\Phi}{\partial x^i}$
$\Gamma^i_{ij}=\Gamma^i_{ji}$	$-\frac{1}{c^2}\frac{\partial\Phi}{\partial x^j}$	$-\frac{1}{c^2}\frac{\partial\Phi}{\partial x^j}$

Figure 28 – Composantes non nulles de la métrique en champ faible et des connexions affines au premier ordre associées. Les différentes composantes sont exprimées en coordonnées spatiales comobiles cartésiennes et en temps cosmique t sur la colonne de gauche tandis qu'elles sont exprimées en temps conforme η sur la colonne de droite. Les termes du second ordre impliquant Ψ, Φ et leurs dérivées respectives sont négligés.

ces expressions, u_μ représente la quadri-vitesse et $g_{\mu\nu}$ est le tenseur métrique perturbé diagonal des paragraphes précédents où l'on ne considèrera qu'un seul potentiel $\Psi = \Phi$. En conséquence, en ne retenant que les perturbations du premier ordre, on peut écrire :

$$\begin{aligned} T_{\mu\nu} &= \left[\left(\bar{\rho} + \delta\rho\right)\rho c^2 + \left(\bar{P} + \delta P\right)\right]\left(\bar{u}_\mu + \delta u_\mu\right)\left(\bar{u}_\nu + \delta u_\nu\right) + \left(\bar{P} + \delta P\right)\left(\bar{g}_{\mu\nu} + \delta g_{\mu\nu}\right) \\ \bar{T}_{\mu\nu} &= \left(\bar{\rho}c^2 + \bar{P}\right)\bar{u}_\mu\bar{u}_\nu + \bar{P}\bar{g}_{\mu\nu} \\ \delta T_{\mu\nu} &= \left(\delta\rho c^2 + \delta P\right)\bar{u}_\mu\bar{u}_\nu + \delta P\bar{g}_{\mu\nu} + \left(\bar{\rho}c^2 + \bar{P}\right)\left(\bar{u}_\mu\delta u_\nu + \delta u_\mu\bar{u}_\nu\right) + \bar{P}\delta g_{\mu\nu} \end{aligned} \tag{3.24}$$

D'autre part, on considère une normalisation de la quadri-vitesse égale à :

$$\boxed{u_\mu u^\mu = -1} \tag{3.25}$$

Or comme :

$$\begin{aligned} \bar{T}_{00} \propto \rho \quad &\Rightarrow \quad \bar{u}_0\bar{u}_0 = -g_{00} = c^2 \\ \bar{T}^{00} \propto \rho \quad &\Rightarrow \quad \bar{u}^0\bar{u}^0 = -g^{00} = \frac{1}{c^2} \end{aligned} \quad \text{et} \quad \begin{aligned} \bar{T}_{ii} \propto P \quad &\Rightarrow \quad \bar{u}_i\bar{u}_i = 0 \\ \bar{T}^{ii} \propto P \quad &\Rightarrow \quad \bar{u}^i\bar{u}^i = 0 \end{aligned} \tag{3.26}$$

on en déduit que la normalisation, valable également sur les termes d'ordre zéro seuls $\bar{u}_\mu\bar{u}^\mu = -1$, implique :

$$\bar{u}^\mu = \frac{1}{c}\delta^\mu_0 \qquad \text{et} \qquad \bar{u}_\mu = -c\delta^0_\mu \tag{3.27}$$

De plus :

$$\begin{aligned} u_\mu u^\mu = g_{\mu\nu}u^\nu u^\mu \quad &= \left(\bar{g}_{\mu\nu} + \delta g_{\mu\nu}\right)\left(\bar{u}^\nu + \delta u^\nu\right)\left(\bar{u}^\mu + \delta u^\mu\right) \approx \bar{u}_\mu\bar{u}^\mu + 2\bar{u}_\mu\delta u^\mu + \delta g_{\mu\nu}\bar{u}^\nu\bar{u}^\mu \\ u_\mu u^\mu = g^{\mu\nu}u_\mu u_\nu \quad &= \left(\bar{g}^{\mu\nu} + \delta g^{\mu\nu}\right)\left(\bar{u}_\mu + \delta u_\mu\right)\left(\bar{u}_\nu + \delta u_\nu\right) \approx \bar{u}_\mu\bar{u}^\mu + 2\bar{u}^\mu\delta u_\mu + \delta g^{\mu\nu}\bar{u}_\mu\bar{u}_\nu \end{aligned} \tag{3.28}$$

si bien que la normalisation impose pour les termes du premier ordre :

$$\begin{aligned} 2\bar{u}_\mu\delta u^\mu + \delta g_{\mu\nu}\bar{u}^\nu\bar{u}^\mu = 0 \quad &\Rightarrow \quad \delta u^\mu = \frac{1}{2}\delta g_{\mu\nu}\bar{u}^\mu\bar{u}^\mu\bar{u}^\nu \\ 2\bar{u}^\mu\delta u_\mu + \delta g^{\mu\nu}\bar{u}_\mu\bar{u}_\nu = 0 \quad &\Rightarrow \quad \delta u_\mu = \frac{1}{2}\delta g^{\mu\nu}\bar{u}_\mu\bar{u}_\mu\bar{u}_\nu \end{aligned} \tag{3.29}$$

soit au final :

$$\boxed{\begin{aligned} \delta u^\mu &= -\frac{1}{2}\bar{g}^{\mu\nu}\delta g_{\mu\nu}\bar{u}^\mu \\ \delta u_\mu &= -\frac{1}{2}\bar{g}_{\mu\nu}\delta g^{\mu\nu}\bar{u}_\mu \end{aligned}} \quad \text{et donc :} \quad \begin{aligned} \delta u^0 &= -\frac{\Phi}{c^3} \\ \delta u_0 &= -\frac{\Phi}{c} \end{aligned} \tag{3.30}$$

En injectant 3.27 et 3.30 dans 3.24, on obtient alors le tenseur énergie-impulsion perturbé donné en figure 29.

3.2.3 Équations d'Einstein perturbées

Le tenseur d'Einstein perturbé et le tenseur énergie-impulsion perturbé ayant été déterminés, il devient possible d'écrire les équations d'Einstein perturbées :

$$\boxed{\bar{G}_{\mu\nu} + \delta G_{\mu\nu} = \frac{8\pi G}{c^4}\left(\bar{T}_{\mu\nu} + \delta T_{\mu\nu}\right)} \tag{3.31}$$

Dans la suite, on pose $\Psi = \Phi$ et :

$$H = \frac{\dot{a}}{a} \qquad \text{et} \qquad \dot{H} = \frac{a\ddot{a} - \dot{a}^2}{a^2} \tag{3.32}$$

Ricci curvature tensor	
R_{00}	$-\frac{3\ddot{a}}{a}+\frac{3\dot{a}}{ac^2}\left(\frac{\partial\Psi}{\partial t}+2\frac{\partial\Phi}{\partial t}\right)+\frac{3}{c^2}\frac{\partial^2\Phi}{\partial t^2}+\frac{1}{a^2}\nabla^2\Psi$
R_{ii}	$\frac{a\ddot{a}}{c^2}+\frac{2\dot{a}^2}{c^2}-\left(\frac{2a\ddot{a}}{c^4}+\frac{4\dot{a}^2}{c^4}\right)(\Psi+\Phi)-\frac{a\dot{a}}{c^4}\left(\frac{\partial\Psi}{\partial t}+6\frac{\partial\Phi}{\partial t}\right)$ $-\frac{a^2}{c^4}\frac{\partial^2\Phi}{\partial t^2}+\frac{1}{c^2}\left(-\frac{\partial^2\Psi}{\partial {x^i}^2}+\frac{\partial^2\Phi}{\partial {x^i}^2}+\nabla^2\Phi\right)$
$R_{0i}=R_{i0}$	$\frac{2\dot{a}}{ac^2}\frac{\partial\Psi}{\partial x^i}+\frac{2}{c^2}\frac{\partial^2\Phi}{\partial x^i\partial t}$
$R_{ij}=R_{ji}$	$-\frac{1}{c^2}\frac{\partial^2\Psi}{\partial x^i\partial x^j}+\frac{1}{c^2}\frac{\partial^2\Phi}{\partial x^i\partial x^j}$

Curvature scalar	
R	$\frac{6\dot{a}^2}{a^2c^2}+\frac{6\ddot{a}}{ac^2}-\frac{12}{c^4}\left(\frac{\dot{a}^2}{a^2}+\frac{\ddot{a}}{a}\right)\Psi-\frac{6\dot{a}}{ac^4}\left(\frac{\partial\Psi}{\partial t}+4\frac{\partial\Phi}{\partial t}\right)$ $-\frac{6}{c^4}\frac{\partial^2\Phi}{\partial t^2}-\frac{2}{a^2c^2}\nabla^2\Psi+\frac{4}{a^2c^2}\nabla^2\Phi$
$\frac{1}{2}g_{00}R$	$-\frac{3\dot{a}^2}{a^2}-\frac{3\ddot{a}}{a}+\frac{3\dot{a}}{ac^2}\left(\frac{\partial\Psi}{\partial t}+4\frac{\partial\Phi}{\partial t}\right)+\frac{3}{c^2}\frac{\partial^2\Phi}{\partial t^2}+\frac{1}{a^2}\nabla^2\Psi-\frac{2}{a^2}\nabla^2\Phi$
$\frac{1}{2}g_{ii}R$	$\frac{3\dot{a}^2}{c^2}+\frac{3a\ddot{a}}{c^2}-\frac{6}{c^4}\left(\dot{a}^2+a\ddot{a}\right)(\Psi+\Phi)-\frac{3a\dot{a}}{c^4}\left(\frac{\partial\Psi}{\partial t}+4\frac{\partial\Phi}{\partial t}\right)$ $-\frac{3a^2}{c^4}\frac{\partial^2\Phi}{\partial t^2}-\frac{1}{c^2}\nabla^2\Psi+\frac{2}{c^2}\nabla^2\Phi$

Einstein tensor	
G_{00}	$\frac{3\dot{a}^2}{a^2}-\frac{6\dot{a}}{ac^2}\frac{\partial\Phi}{\partial t}+\frac{2}{a^2}\nabla^2\Phi$
G_{ii}	$-\frac{2a\ddot{a}}{c^2}-\frac{\dot{a}^2}{c^2}+\frac{2}{c^4}\left(\dot{a}^2+2a\ddot{a}\right)(\Psi+\Phi)+\frac{2a\dot{a}}{c^4}\left(\frac{\partial\Psi}{\partial t}+3\frac{\partial\Phi}{\partial t}\right)$ $+\frac{2a^2}{c^4}\frac{\partial^2\Phi}{\partial t^2}+\frac{1}{c^2}\left(\nabla^2\Psi-\frac{\partial^2\Psi}{\partial {x^i}^2}-\nabla^2\Phi+\frac{\partial^2\Phi}{\partial {x^i}^2}\right)$
$G_{0i}=G_{i0}$	$\frac{2\dot{a}}{ac^2}\frac{\partial\Psi}{\partial x^i}+\frac{2}{c^2}\frac{\partial^2\Phi}{\partial x^i\partial t}$
$G_{ij}=G_{ji}$	$-\frac{1}{c^2}\frac{\partial^2\Psi}{\partial x^i\partial x^j}+\frac{1}{c^2}\frac{\partial^2\Phi}{\partial x^i\partial x^j}$

Stress–energy tensor	
T_{00}	$\bar{\rho}c^4\left(1+\frac{\delta\rho}{\bar{\rho}}+2\frac{\Phi}{c^2}\right)$
T_{ii}	$\bar{P}a^2\left(1+\frac{\delta P}{\bar{P}}-2\frac{\Phi}{c^2}\right)$
$T_{0i}=T_{i0}$	$-\bar{\rho}c^3\delta u_i\left(1+\frac{\bar{P}}{\bar{\rho}c^2}\right)$
$T_{ij}=T_{ji}$	0

Figure 29 – Composantes des tenseurs de l'équation d'Einstein en champ faible. Le tenseur de Ricci est obtenu comme une contraction du tenseur de Riemann. Le tenseur d'Einstein est calculé à partir du tenseur de Ricci, de la métrique et du scalaire de courbure. Pour le tenseur énergie-impulsion, on considère un seul potentiel $\Psi=\Phi$. ∇^2 est le Laplacien : $\Delta\Phi=\nabla^2\Phi=\frac{\partial^2\Phi}{\partial x^2}+\frac{\partial^2\Phi}{\partial y^2}+\frac{\partial^2\Phi}{\partial z^2}$.

Les composantes spatiales $G_{ii} = \frac{8\pi G}{c^4} T_{ii}$ s'écrivent :

$$-\dot{a}^2 - 2a\ddot{a} + \frac{4}{c^2}\left(\dot{a}^2 + 2a\ddot{a}\right)\Phi + \frac{8a\dot{a}}{c^2}\frac{\partial\Phi}{\partial t} + \frac{2a^2}{c^2}\frac{\partial^2\Phi}{\partial t^2} = \frac{8\pi G}{c^2}\bar{P}a^2\left(1 + \frac{\delta P}{\bar{P}} - 2\frac{\Phi}{c^2}\right) \tag{3.33}$$

L'ordre zéro donne la seconde équation de Friedmann-Lemaître :

$$\boxed{-\frac{\dot{a}^2}{a^2} - 2\frac{\ddot{a}}{a} = \frac{8\pi G}{c^2}\bar{P}} \quad \Leftrightarrow \quad -3H^2 - 2\dot{H} = \frac{8\pi G}{c^2}\bar{P} \tag{3.34}$$

et le premier ordre conduit à :

$$\boxed{2\left(\dot{a}^2 + 2a\ddot{a}\right)\Phi + 4a\dot{a}\frac{\partial\Phi}{\partial t} + a^2\frac{\partial^2\Phi}{\partial t} = 4\pi G a^2\delta P - \frac{8\pi G}{c^2}a^2\Phi\bar{P}} \tag{3.35}$$

La composante temporelle $G_{00} = \frac{8\pi G}{c^4} T_{00}$ s'écrit :

$$\frac{3\dot{a}^2}{a^2} - \frac{6\dot{a}}{ac^2}\frac{\partial\Phi}{\partial t} + \frac{2}{a^2}\nabla^2\Phi = 8\pi G\bar{\rho}\left(1 + \frac{\delta\rho}{\bar{\rho}} + 2\frac{\Phi}{c^2}\right) \tag{3.36}$$

L'ordre zéro donne la première équation de Friedmann-Lemaître :

$$\boxed{3\frac{\dot{a}^2}{a^2} = 8\pi G\bar{\rho}} \quad \Leftrightarrow \quad 3H^2 = 8\pi G\bar{\rho} \tag{3.37}$$

tandis que les termes du premier ordre donnent :

$$\nabla^2\Phi = 4\pi G a^2\delta\rho + 8\pi G a^2\bar{\rho}\frac{\Phi}{c^2} + \frac{3a\dot{a}}{c^2}\frac{\partial\Phi}{\partial t} \tag{3.38}$$

En simplifiant le membre de droite par 3.37, on obtient :

$$\boxed{\nabla^2\Phi = 4\pi G a^2\delta\rho + 3\dot{a}\left(\frac{\dot{a}}{c^2}\Phi + \frac{a}{c^2}\frac{\partial\Phi}{\partial t}\right)} \tag{3.39}$$

Finalement, les termes mixes $G_{0i} = \frac{8\pi G}{c^4} T_{0i}$, eux, sont uniquement du premier ordre et s'écrivent :

$$\frac{\partial}{\partial x^i}\left(\frac{\dot{a}}{c^2}\Phi + \frac{a}{c^2}\frac{\partial\Phi}{\partial t}\right) = -\frac{4\pi G}{c^3}a\left(\bar{\rho}c^2 + \bar{P}\right)\delta u_i \tag{3.40}$$

Se pose alors la question de la nature de la perturbation δu_i. Si l'on considère la partie scalaire de la décomposition scalaire-vecteur-tenseur du potentiel de 3-vitesse v associé à cette perturbation, on peut écrire la perturbation comme :

$$\delta u_i = \partial_i v \tag{3.41}$$

Par intégration de l'équation précédente en supposant que les dérivées des perturbations sont nulles à l'infini, on obtient donc :

$$\frac{\dot{a}}{c^2}\Phi + \frac{a}{c^2}\frac{\partial\Phi}{\partial t} = -\frac{4\pi G}{c^3}a\left(\bar{\rho}c^2 + \bar{P}\right)v \tag{3.42}$$

ce qui, injecté dans 3.39 donne finalement l'équation de Poisson :

$$\boxed{\nabla^2\Phi = 4\pi G a^2\bar{\rho}\left[\frac{\delta\rho}{\bar{\rho}} - 3\frac{\dot{a}v}{ac}\left(1 + \frac{\bar{P}}{\bar{\rho}c^2}\right)\right] = 4\pi G a^2\Delta} \quad \text{avec} \quad \boxed{\Delta = \delta\rho - 3\frac{\dot{a}v}{ac}\left(\bar{\rho} + \frac{\bar{P}}{c^2}\right)} \tag{3.43}$$

où $\dot{a}v$ a la dimension d'une vitesse et où Δ est le facteur de surdensité dans la jauge comobile, jauge pour laquelle il y a annulation des perturbations de vitesse $v = B = 0$[102].

3.2.4 Quantités calculées dans les simulations cosmologiques

Pour faire évoluer la distribution de matière dans un univers en expansion, les codes dynamiques de simulations cosmologiques résolvent l'équation de Poisson en coordonnées comobiles :

$$\nabla^2 \Phi_{\text{sim}} = 4\pi G a^2 \delta\rho_{\text{sim}} \tag{3.44}$$

Toutefois, une grande attention doit être portée à la normalisation des différentes variables. Les calculs précédents ont permis de s'assurer de la compatibilité du potentiel calculé dans les simulations avec le potentiel intervenant dans la métrique en champ faible avec $\Psi = \Phi$ et dans le cas où l'on a négligé tous les termes du second ordre sur Φ et ses dérivées, y compris ceux du type $\Phi\nabla^2\Phi$. De plus, l'équation liant le potentiel aux surdensités 3.43 obtenue par résolution des équations d'Einstein en champ faible est compatible avec l'équation de Poisson sur Φ_{sim} et $\delta\rho_{\text{sim}}$ manipulée par les simulations cosmologiques, ces dernières travaillant en jauge comobile et non en jauge newtonienne[103].

Dans ces conditions, Φ_{sim} peut donc être directement identifié avec le potentiel Φ des perturbations de la métrique 3.15. Ce potentiel est relié à la surdensité dans la jauge newtonienne par l'équation 3.43 et à la surdensité dans la jauge comobile par l'équation 3.44. Dans les deux cas, les dérivées spatiales ∇^2 sont des dérivées par rapport aux coordonnées comobiles, donc dont le calcul ne fait pas intervenir de facteur a dans le cas d'une boîte de simulation dont le côté est normalisé à 1 pour tout temps cosmique t, tandis que ρ est une densité physique, et donc d'autant plus grande que t est petit pour un univers en expansion. Dans le cas de simulations basées sur des coordonnées supercomobiles[61], comme c'est le cas pour les simulations utilisant le code RAMSES, les conversions d'unités adéquates devront être faites pour se replacer en coordonnées comobiles classiques, là où les correspondances avec la métrique de champ faible ont été établies.

3.3 Établissement des équations géodésiques sous-tendant la propagation de la lumière

3.3.1 Cas général

Maintenant que la correspondance entre les quantités physiques calculées dans les simulations numériques et celles figurant dans la métrique de champ faible a été établie, concentrons-nous sur l'obtention des équations géodésiques sur lesquelles reposera le traitement de la propagation de la lumière. Sous leur forme la plus générale, elles s'écrivent :

$$\boxed{\frac{\mathrm{d}^2 x^\alpha}{\mathrm{d}\lambda^2} = -\Gamma^\alpha_{\beta\gamma} \frac{\mathrm{d}x^\beta}{\mathrm{d}\lambda} \frac{\mathrm{d}x^\gamma}{\mathrm{d}\lambda}} \tag{3.45}$$

où les $\Gamma^\alpha_{\beta\gamma}$ sont les symboles de Christoffel pour la métrique considérée et où λ est le paramètre affine permettant de spécifier la coordonnée curviligne le long de la géodésique. En injectant les connexions affines de la figure 28 dans cette équation, on obtient alors un système de quatre équations différentielles du second ordre paramétrées par λ et décrivant la forme de la géodésique. Les équations obtenues sont présentées en coordonnées cartésiennes en figure 30 et en coordonnées sphériques en figure 31. Les expressions sont valables y compris pour $\mathrm{d}s^2 \neq 0$, c'est-à-dire pour des particules massives. Pour une telle particule, la connaissance de Ψ, Φ et de leurs dérivées premières spatiales et temporelles permet d'intégrer le système et donc d'accéder à la trajectoire dans l'espace-temps. Comme attendu, à partir de la distribution d'un potentiel arbitraire, extrait de simulations numériques ou reconstruit à partir de données observationnelles, ce système d'équations permet de traiter de façon générique la propagation de la lumière dans un univers structuré sans qu'il y ait besoin de faire intervenir des approximations de nature géométrique lié à un formalisme particulier.

Geodesics equations in cartesian comoving coordinates	
$\mathbf{d}s^2$	$-c^2\left(1+2\frac{\Psi}{c^2}\right)\mathrm{d}t^2+a^2\left(1-2\frac{\Phi}{c^2}\right)\left(\mathrm{d}x^2+\mathrm{d}y^2+\mathrm{d}z^2\right)$
$\frac{\mathbf{d}^2t}{\mathbf{d}\lambda^2}$	$-\frac{1}{c^2}\frac{\partial\Psi}{\partial t}\left(\frac{\mathrm{d}t}{\mathrm{d}\lambda}\right)^2-\frac{2}{c^2}\frac{\mathrm{d}t}{\mathrm{d}\lambda}\left[\frac{\partial\Psi}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}\lambda}+\frac{\partial\Psi}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}\lambda}+\frac{\partial\Psi}{\partial z}\frac{\mathrm{d}z}{\mathrm{d}\lambda}\right]$ $-\left[\frac{a\dot{a}}{c^2}-\frac{2a\dot{a}}{c^4}(\Psi+\Phi)-\frac{a^2}{c^4}\frac{\partial\Phi}{\partial t}\right]\left[\left(\frac{\mathrm{d}x}{\mathrm{d}\lambda}\right)^2+\left(\frac{\mathrm{d}y}{\mathrm{d}\lambda}\right)^2+\left(\frac{\mathrm{d}z}{\mathrm{d}\lambda}\right)^2\right]$
$\frac{\mathbf{d}^2x}{\mathbf{d}\lambda^2}$	$-\frac{1}{a^2}\frac{\partial\Psi}{\partial x}\left(\frac{\mathrm{d}t}{\mathrm{d}\lambda}\right)^2-2\left(\frac{\dot{a}}{a}-\frac{1}{c^2}\frac{\partial\Phi}{\partial t}\right)\frac{\mathrm{d}t}{\mathrm{d}\lambda}\frac{\mathrm{d}x}{\mathrm{d}\lambda}+\frac{2}{c^2}\frac{\mathrm{d}x}{\mathrm{d}\lambda}\left[\frac{\partial\Phi}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}\lambda}+\frac{\partial\Phi}{\partial z}\frac{\mathrm{d}z}{\mathrm{d}\lambda}\right]$ $+\frac{1}{c^2}\frac{\partial\Phi}{\partial x}\left[\left(\frac{\mathrm{d}x}{\mathrm{d}\lambda}\right)^2-\left(\frac{\mathrm{d}y}{\mathrm{d}\lambda}\right)^2-\left(\frac{\mathrm{d}z}{\mathrm{d}\lambda}\right)^2\right]$
$\frac{\mathbf{d}^2y}{\mathbf{d}\lambda^2}$	$-\frac{1}{a^2}\frac{\partial\Psi}{\partial y}\left(\frac{\mathrm{d}t}{\mathrm{d}\lambda}\right)^2-2\left(\frac{\dot{a}}{a}-\frac{1}{c^2}\frac{\partial\Phi}{\partial t}\right)\frac{\mathrm{d}t}{\mathrm{d}\lambda}\frac{\mathrm{d}y}{\mathrm{d}\lambda}+\frac{2}{c^2}\frac{\mathrm{d}y}{\mathrm{d}\lambda}\left[\frac{\partial\Phi}{\partial z}\frac{\mathrm{d}z}{\mathrm{d}\lambda}+\frac{\partial\Phi}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}\lambda}\right]$ $+\frac{1}{c^2}\frac{\partial\Phi}{\partial y}\left[\left(\frac{\mathrm{d}y}{\mathrm{d}\lambda}\right)^2-\left(\frac{\mathrm{d}z}{\mathrm{d}\lambda}\right)^2-\left(\frac{\mathrm{d}x}{\mathrm{d}\lambda}\right)^2\right]$
$\frac{\mathbf{d}^2z}{\mathbf{d}\lambda^2}$	$-\frac{1}{a^2}\frac{\partial\Psi}{\partial z}\left(\frac{\mathrm{d}t}{\mathrm{d}\lambda}\right)^2-2\left(\frac{\dot{a}}{a}-\frac{1}{c^2}\frac{\partial\Phi}{\partial t}\right)\frac{\mathrm{d}t}{\mathrm{d}\lambda}\frac{\mathrm{d}z}{\mathrm{d}\lambda}+\frac{2}{c^2}\frac{\mathrm{d}z}{\mathrm{d}\lambda}\left[\frac{\partial\Phi}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}\lambda}+\frac{\partial\Phi}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}\lambda}\right]$ $+\frac{1}{c^2}\frac{\partial\Phi}{\partial z}\left[\left(\frac{\mathrm{d}z}{\mathrm{d}\lambda}\right)^2-\left(\frac{\mathrm{d}x}{\mathrm{d}\lambda}\right)^2-\left(\frac{\mathrm{d}y}{\mathrm{d}\lambda}\right)^2\right]$

Geodesics equations in cartesian conformal coordinates	
$\mathbf{d}s^2$	$a^2\left[-c^2\left(1+2\frac{\Psi}{c^2}\right)\mathrm{d}\eta^2+\left(1-2\frac{\Phi}{c^2}\right)\left(\mathrm{d}x^2+\mathrm{d}y^2+\mathrm{d}z^2\right)\right]$
$\frac{\mathbf{d}^2\eta}{\mathbf{d}\lambda^2}$	$-\left(\frac{a'}{a}+\frac{1}{c^2}\frac{\partial\Psi}{\partial\eta}\right)\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2-\frac{2}{c^2}\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\left[\frac{\partial\Psi}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}\lambda}+\frac{\partial\Psi}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}\lambda}+\frac{\partial\Psi}{\partial z}\frac{\mathrm{d}z}{\mathrm{d}\lambda}\right]$ $-\left[\frac{a'}{ac^2}-\frac{2a'}{ac^4}(\Psi+\Phi)-\frac{1}{c^4}\frac{\partial\Phi}{\partial\eta}\right]\left[\left(\frac{\mathrm{d}x}{\mathrm{d}\lambda}\right)^2+\left(\frac{\mathrm{d}y}{\mathrm{d}\lambda}\right)^2+\left(\frac{\mathrm{d}z}{\mathrm{d}\lambda}\right)^2\right]$
$\frac{\mathbf{d}^2x}{\mathbf{d}\lambda^2}$	$-\frac{\partial\Psi}{\partial x}\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2-2\left(\frac{a'}{a}-\frac{1}{c^2}\frac{\partial\Phi}{\partial\eta}\right)\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\frac{\mathrm{d}x}{\mathrm{d}\lambda}+\frac{2}{c^2}\frac{\mathrm{d}x}{\mathrm{d}\lambda}\left[\frac{\partial\Phi}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}\lambda}+\frac{\partial\Phi}{\partial z}\frac{\mathrm{d}z}{\mathrm{d}\lambda}\right]$ $+\frac{1}{c^2}\frac{\partial\Phi}{\partial x}\left[\left(\frac{\mathrm{d}x}{\mathrm{d}\lambda}\right)^2-\left(\frac{\mathrm{d}y}{\mathrm{d}\lambda}\right)^2-\left(\frac{\mathrm{d}z}{\mathrm{d}\lambda}\right)^2\right]$
$\frac{\mathbf{d}^2y}{\mathbf{d}\lambda^2}$	$-\frac{\partial\Psi}{\partial y}\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2-2\left(\frac{a'}{a}-\frac{1}{c^2}\frac{\partial\Phi}{\partial\eta}\right)\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\frac{\mathrm{d}y}{\mathrm{d}\lambda}+\frac{2}{c^2}\frac{\mathrm{d}y}{\mathrm{d}\lambda}\left[\frac{\partial\Phi}{\partial z}\frac{\mathrm{d}z}{\mathrm{d}\lambda}+\frac{\partial\Phi}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}\lambda}\right]$ $+\frac{1}{c^2}\frac{\partial\Phi}{\partial y}\left[\left(\frac{\mathrm{d}y}{\mathrm{d}\lambda}\right)^2-\left(\frac{\mathrm{d}z}{\mathrm{d}\lambda}\right)^2-\left(\frac{\mathrm{d}x}{\mathrm{d}\lambda}\right)^2\right]$
$\frac{\mathbf{d}^2z}{\mathbf{d}\lambda^2}$	$-\frac{\partial\Psi}{\partial z}\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2-2\left(\frac{a'}{a}-\frac{1}{c^2}\frac{\partial\Phi}{\partial\eta}\right)\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\frac{\mathrm{d}z}{\mathrm{d}\lambda}+\frac{2}{c^2}\frac{\mathrm{d}z}{\mathrm{d}\lambda}\left[\frac{\partial\Phi}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}\lambda}+\frac{\partial\Phi}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}\lambda}\right]$ $+\frac{1}{c^2}\frac{\partial\Phi}{\partial z}\left[\left(\frac{\mathrm{d}z}{\mathrm{d}\lambda}\right)^2-\left(\frac{\mathrm{d}x}{\mathrm{d}\lambda}\right)^2-\left(\frac{\mathrm{d}y}{\mathrm{d}\lambda}\right)^2\right]$

Figure 30 – Équations géodésiques en coordonnées cartésiennes pour une métrique de champ faible. À partir de la métrique en champ faible, il est possible d'obtenir les équations géodésiques en temps cosmique et en temps conforme présentées ici, liant les différentes coordonnées entre elles le long de la trajectoire curviligne paramétrée par le paramètre affine λ.

Geodesics equations in spherical comoving coordinates	
$\mathbf{d}s^2$	$-c^2\left(1+2\frac{\Psi}{c^2}\right)\mathrm{d}t^2+a^2\left(1-2\frac{\Phi}{c^2}\right)\left(\mathrm{d}r^2+r^2\,\mathrm{d}\theta^2+r^2\sin^2\theta\,\mathrm{d}\phi^2\right)$
$\frac{\mathbf{d}^2t}{\mathbf{d}\lambda^2}$	$-\frac{1}{c^2}\frac{\partial\Psi}{\partial t}\left(\frac{\mathrm{d}t}{\mathrm{d}\lambda}\right)^2-\frac{2}{c^2}\frac{\mathrm{d}t}{\mathrm{d}\lambda}\left[\frac{\partial\Psi}{\partial r}\frac{\mathrm{d}r}{\mathrm{d}\lambda}+\frac{\partial\Psi}{\partial\theta}\frac{\mathrm{d}\theta}{\mathrm{d}\lambda}+\frac{\partial\Psi}{\partial\varphi}\frac{\mathrm{d}\varphi}{\mathrm{d}\lambda}\right]$ $-\left[\frac{a\dot{a}}{c^2}-\frac{2a\dot{a}}{c^4}(\Psi+\Phi)-\frac{a^2}{c^4}\frac{\partial\Phi}{\partial t}\right]\left[\left(\frac{\mathrm{d}r}{\mathrm{d}\lambda}\right)^2+r^2\left(\frac{\mathrm{d}\theta}{\mathrm{d}\lambda}\right)^2+r^2\sin^2\theta\left(\frac{\mathrm{d}\varphi}{\mathrm{d}\lambda}\right)^2\right]$
$\frac{\mathbf{d}^2r}{\mathbf{d}\lambda^2}$	$-\frac{1}{a^2}\frac{\partial\Psi}{\partial r}\left(\frac{\mathrm{d}t}{\mathrm{d}\lambda}\right)^2+\frac{1}{c^2}\frac{\partial\Phi}{\partial r}\left[\left(\frac{\mathrm{d}r}{\mathrm{d}\lambda}\right)^2-r^2\left(\frac{\mathrm{d}\theta}{\mathrm{d}\lambda}\right)^2-r^2\sin^2\theta\left(\frac{\mathrm{d}\varphi}{\mathrm{d}\lambda}\right)^2\right]$ $+r\left(\frac{\mathrm{d}\theta}{\mathrm{d}\lambda}\right)^2+r\sin^2\theta\left(\frac{\mathrm{d}\varphi}{\mathrm{d}\lambda}\right)^2$ $-2\frac{\mathrm{d}r}{\mathrm{d}\lambda}\left[\left(\frac{\dot{a}}{a}-\frac{1}{c^2}\frac{\partial\Phi}{\partial t}\right)\frac{\mathrm{d}t}{\mathrm{d}\lambda}-\frac{1}{c^2}\frac{\partial\Phi}{\partial\theta}\frac{\mathrm{d}\theta}{\mathrm{d}\lambda}-\frac{1}{c^2}\frac{\partial\Phi}{\partial\varphi}\frac{\mathrm{d}\varphi}{\mathrm{d}\lambda}\right]$
$\frac{\mathbf{d}^2\theta}{\mathbf{d}\lambda^2}$	$-\frac{1}{a^2r^2}\frac{\partial\Psi}{\partial\theta}\left(\frac{\mathrm{d}t}{\mathrm{d}\lambda}\right)^2+\frac{1}{c^2}\frac{\partial\Phi}{\partial\theta}\left[\left(\frac{\mathrm{d}\theta}{\mathrm{d}\lambda}\right)^2-\sin^2\theta\left(\frac{\mathrm{d}\varphi}{\mathrm{d}\lambda}\right)^2-\frac{1}{r^2}\left(\frac{\mathrm{d}r}{\mathrm{d}\lambda}\right)^2\right]$ $+\frac{\sin 2\theta}{2}\left(\frac{\mathrm{d}\varphi}{\mathrm{d}\lambda}\right)^2-2\frac{\mathrm{d}\theta}{\mathrm{d}\lambda}\left[\left(\frac{\dot{a}}{a}-\frac{1}{c^2}\frac{\partial\Phi}{\partial t}\right)\frac{\mathrm{d}t}{\mathrm{d}\lambda}+\left(\frac{1}{r}-\frac{1}{c^2}\frac{\partial\Phi}{\partial r}\right)\frac{\mathrm{d}r}{\mathrm{d}\lambda}-\frac{1}{c^2}\frac{\partial\Phi}{\partial\varphi}\frac{\mathrm{d}\varphi}{\mathrm{d}\lambda}\right]$
$\frac{\mathbf{d}^2\varphi}{\mathbf{d}\lambda^2}$	$-\frac{1}{a^2r^2\sin^2\theta}\frac{\partial\Psi}{\partial\varphi}\left(\frac{\mathrm{d}t}{\mathrm{d}\lambda}\right)^2+\frac{1}{c^2}\frac{\partial\Phi}{\partial\varphi}\left[\left(\frac{\mathrm{d}\varphi}{\mathrm{d}\lambda}\right)^2-\frac{1}{r^2\sin^2\theta}\left(\frac{\mathrm{d}r}{\mathrm{d}\lambda}\right)^2-\frac{1}{r^2}\left(\frac{\mathrm{d}\theta}{\mathrm{d}\lambda}\right)^2\right]$ $-2\frac{\mathrm{d}\varphi}{\mathrm{d}\lambda}\left[\left(\frac{\dot{a}}{a}-\frac{1}{c^2}\frac{\partial\Phi}{\partial t}\right)\frac{\mathrm{d}t}{\mathrm{d}\lambda}+\left(\frac{1}{r}-\frac{1}{c^2}\frac{\partial\Phi}{\partial r}\right)\frac{\mathrm{d}r}{\mathrm{d}\lambda}+\left(\frac{\cos\theta}{\sin\theta}-\frac{1}{c^2}\frac{\partial\Phi}{\partial\theta}\right)\frac{\mathrm{d}\theta}{\mathrm{d}\lambda}\right]$

Figure 31 – Équations géodésiques en coordonnées sphériques pour une métrique de champ faible. Dans certains cas, il peut être utile de se placer en coordonnées sphériques pour étudier analytiquement la propagation de la lumière autour d'un observateur central. Est donné ci-dessus le système d'équations équivalent à ceux de la figure 30 pour de telles coordonnées.

3.3.2 Cas des géodésiques nulles

Dans le cas de particules de masse nulle le système d'équations de la figure 30 peut être simplifié et mis sous une forme davantage compatible avec les simulations numériques à disposition dans la suite. Cette réécriture est motivée avant tout par des critères de performance numérique, mais le système de la figure 30 en temps conforme pourrait être implémenté tel quel et fournirait le même résultat final.

Dans toute la suite, on suppose $\Psi = \Phi$ et on note $\mathrm{d}\Sigma^2 = \mathrm{d}x^2 + \mathrm{d}y^2 + \mathrm{d}z^2$. En réarrangeant les termes du système en temps conforme, on peut écrire :

$$\begin{aligned}
\frac{\mathrm{d}^2\eta}{\mathrm{d}\lambda^2} =& -\frac{a'}{ac^2}\left(1-2\frac{\Phi}{c^2}\right)\left[-c^2\left(1+2\frac{\Phi}{c^2}\right)\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2+\left(1-2\frac{\Phi}{c^2}\right)\left(\frac{\mathrm{d}\Sigma}{\mathrm{d}\lambda}\right)^2\right] \\
&-\frac{2a'}{a}\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2-\frac{2}{c^2}\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\left[\frac{\partial\Phi}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}\lambda}+\frac{\partial\Phi}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}\lambda}+\frac{\partial\Phi}{\partial z}\frac{\mathrm{d}z}{\mathrm{d}\lambda}\right] \\
&+\frac{1}{c^4}\frac{\partial\Phi}{\partial\eta}\left[-c^2\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2+\left(\frac{\mathrm{d}\Sigma}{\mathrm{d}\lambda}\right)^2\right] \\
\frac{\mathrm{d}^2x}{\mathrm{d}\lambda^2} =& -\frac{2a'}{a}\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\frac{\mathrm{d}x}{\mathrm{d}\lambda}+\frac{2}{c^2}\frac{\mathrm{d}\Phi}{\mathrm{d}\lambda}\frac{\mathrm{d}x}{\mathrm{d}\lambda}-2\frac{\partial\Phi}{\partial x}\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2-\frac{1}{c^2}\frac{\partial\Phi}{\partial x}\left[-c^2\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2+\left(\frac{\mathrm{d}\Sigma}{\mathrm{d}\lambda}\right)^2\right]
\end{aligned} \tag{3.46}$$

or :

$$-c^2\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2+\left(\frac{\mathrm{d}\Sigma}{\mathrm{d}\lambda}\right)^2 = \mathrm{d}s^2+2\frac{\Phi}{c^2}\left[c^2\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2+\left(\frac{\mathrm{d}\Sigma}{\mathrm{d}\lambda}\right)^2\right] \tag{3.47}$$

si bien que dans le cas de géodésiques nulles, $\mathrm{d}s^2 = 0$ implique que ce terme est uniquement du premier ordre en Φ. Ainsi, en ne gardant que les termes du premier ordre dans le système d'équations 3.46 devient :

$$\boxed{\begin{aligned}
\frac{d^2\eta}{d\lambda^2} &= -\frac{2a'}{a}\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2-\frac{2}{c^2}\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\left(\frac{\partial\Phi}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}\lambda}+\frac{\partial\Phi}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}\lambda}+\frac{\partial\Phi}{\partial z}\frac{\mathrm{d}z}{\mathrm{d}\lambda}\right) \\
&= -\frac{2a'}{a}\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}-\frac{2}{c^2}\frac{\mathrm{d}\Phi}{\mathrm{d}\lambda}\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}+2\frac{\partial\Phi}{\partial\eta}\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2 \\
\frac{d^2x}{d\lambda^2} &= -\frac{2a'}{a}\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\frac{\mathrm{d}x}{\mathrm{d}\lambda}+\frac{2}{c^2}\frac{\mathrm{d}\Phi}{\mathrm{d}\lambda}\frac{\mathrm{d}x}{\mathrm{d}\lambda}-2\frac{\partial\Phi}{\partial x}\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2 \\
\frac{d^2y}{d\lambda^2} &= -\frac{2a'}{a}\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\frac{\mathrm{d}y}{\mathrm{d}\lambda}+\frac{2}{c^2}\frac{\mathrm{d}\Phi}{\mathrm{d}\lambda}\frac{\mathrm{d}y}{\mathrm{d}\lambda}-2\frac{\partial\Phi}{\partial y}\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2 \\
\frac{d^2z}{d\lambda^2} &= -\frac{2a'}{a}\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\frac{\mathrm{d}z}{\mathrm{d}\lambda}+\frac{2}{c^2}\frac{\mathrm{d}\Phi}{\mathrm{d}\lambda}\frac{\mathrm{d}z}{\mathrm{d}\lambda}-2\frac{\partial\Phi}{\partial z}\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2
\end{aligned}} \tag{3.48}$$

avec :

$$\frac{\mathrm{d}\Phi}{\mathrm{d}\lambda} = \frac{\partial\Phi}{\partial\eta}\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}+\frac{\partial\Phi}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}\lambda}+\frac{\partial\Phi}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}\lambda}+\frac{\partial\Phi}{\partial z}\frac{\mathrm{d}z}{\mathrm{d}\lambda} \tag{3.49}$$

et où la première forme de $\frac{d^2\eta}{d\lambda^2}$ présente l'avantage du point de vue de son implémentation de ne dépendre que de la dérivée totale et de la dérivée spatiale de Φ, sans faire intervenir directement sa dérivée temporelle.

3.3.3 Formulation alternative

Cherchons maintenant à relier le système 3.46 à des expressions plus connues dans le formalisme du lentillage. On a :

$$\frac{\mathrm{d}^2x^i}{\mathrm{d}\lambda^2} = \frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\frac{\mathrm{d}}{\mathrm{d}\eta}\left(\frac{\mathrm{d}x^i}{\mathrm{d}\eta}\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right) = \left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2\frac{\mathrm{d}^2x^i}{\mathrm{d}\eta^2}+\frac{\mathrm{d}x^i}{\mathrm{d}\eta}\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\frac{\mathrm{d}}{\mathrm{d}\eta}\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right) = \left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2\frac{\mathrm{d}^2x^i}{\mathrm{d}\eta^2}+\frac{\mathrm{d}x^i}{\mathrm{d}\eta}\frac{\mathrm{d}^2\eta}{\mathrm{d}\lambda^2} \tag{3.50}$$

donc 3.46 peut se réécrire sous la forme :

$$\frac{\mathrm{d}^2 x^i}{\mathrm{d}\eta^2} = \frac{2}{c^2}\frac{\mathrm{d}\Phi}{\mathrm{d}\eta}\frac{\mathrm{d}x^i}{\mathrm{d}\eta} - 2\frac{\partial\Phi}{\partial x^i} + \frac{2}{c^2}\frac{\mathrm{d}x^i}{\mathrm{d}\eta}\left(\frac{\partial\Phi}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}\eta} + \frac{\partial\Phi}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}\eta} + \frac{\partial\Phi}{\partial z}\frac{\mathrm{d}z}{\mathrm{d}\eta}\right) \tag{3.51}$$

ce qui permet d'obtenir :

$$\vec{w} = w\vec{n} = \frac{\mathrm{d}x}{\mathrm{d}\eta}\vec{e}_x + \frac{\mathrm{d}y}{\mathrm{d}\eta}\vec{e}_y + \frac{\mathrm{d}z}{\mathrm{d}\eta}\vec{e}_z \quad \Rightarrow \quad \frac{\mathrm{d}\vec{w}}{\mathrm{d}\eta} = \frac{2}{c^2}\frac{\mathrm{d}\Phi}{\mathrm{d}\eta}\vec{w} - 2\left[\vec{\nabla}\Phi - \left(\vec{\nabla}\Phi\cdot\vec{n}\right)\frac{w^2}{c^2}\vec{n}\right] \tag{3.52}$$

où $\vec{n}$ est la direction de propagation normalisée : $\|\vec{n}\| = 1$. Or :

$$\mathrm{d}s^2 = 0 \quad \Rightarrow \quad -c^2\left(1+2\frac{\Phi}{c^2}\right) + \left(1-2\frac{\Phi}{c^2}\right)w^2 = 0 \quad \Rightarrow \quad w^2 \approx c^2\left(1+4\frac{\Phi}{c^2}\right) \tag{3.53}$$

donc au premier ordre :

$$\frac{\mathrm{d}\vec{w}}{\mathrm{d}\eta} = \frac{2}{c^2}\frac{\mathrm{d}\Phi}{\mathrm{d}\eta}\vec{w} - 2\left[\vec{\nabla}\Phi - \vec{\nabla}_{\parallel}\Phi - 4\frac{\Phi}{c^2}\left(\vec{\nabla}\Phi\cdot\vec{n}\right)\vec{n}\right] \approx \frac{2}{c^2}\frac{\mathrm{d}\Phi}{\mathrm{d}\eta}\vec{w} - 2\vec{\nabla}_{\perp}\Phi \tag{3.54}$$

ce qui permet de retrouver[94] :

$$\boxed{\frac{\mathrm{d}\vec{n}}{\mathrm{d}\eta} \approx -\frac{2}{c}\vec{\nabla}_{\perp}\Phi} \tag{3.55}$$

Cette dernière équation, qui est centrale pour traiter des problématiques de lentillage, permet de lier directement la variation de direction de rayons lumineux au gradient du potentiel dans le plan perpendiculaire à la propagation. Une approche encore différente du problème consisterait à utiliser les équations de Sachs[104]. Toutefois, la volonté de proposer un traitement unifié aussi générique que possible demande de ne pas dépendre d'un formalisme particulier, et demande donc de rester aussi proche que possible de la formulation relativiste des équations géodésiques. C'est pourquoi la forme paramétrique générale obtenue en figure 30 sera privilégiée dans la suite des traitements.

3.3.4 Expression du redshift

Les sections précédentes ont permis de mettre en évidence les équations paramétriques satisfaites par la trajectoire d'un photon dans un univers faiblement perturbé. La dernière quantité nécessaire aux analyses présentées en figure 27 est le redshift. Dans le cas le plus général, il s'exprime sous la forme 3.1. Or, pour la métrique en coordonnées conformes 3.15, et les perturbations du tenseur énergie-impulsion 3.30, on a :

$$g_{\mu\nu}k^\mu u^\nu = -a^2c^2\left(1+2\frac{\Phi}{c^2}\right)k^0\left(\frac{1}{c}-\frac{\Phi}{c^3}\right) + a^2\left(1-2\frac{\Phi}{c^2}\right)\left(k^1u^1+k^2u^2+k^3u^3\right) \tag{3.56}$$

si bien que le redshift est entièrement spécifié par :

$$\boxed{z = \frac{\left(g_{\mu\nu}k^\mu u^\nu\right)_S}{\left(g_{\mu\nu}k^\mu u^\nu\right)_O} - 1} \quad \text{avec} \quad \boxed{g_{\mu\nu}k^\mu u^\nu = -a^2ck^0\left(1+\frac{\Phi}{c^2}-\frac{k^iu^i}{ck^0}\right)} \tag{3.57}$$

où :

$$\left(k^0,k^1,k^2,k^3\right) = \left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda},\frac{\mathrm{d}x}{\mathrm{d}\lambda},\frac{\mathrm{d}y}{\mathrm{d}\lambda},\frac{\mathrm{d}z}{\mathrm{d}\lambda}\right) \tag{3.58}$$

Connaissant le potentiel à l'emplacement de l'observateur Φ_O et celui à l'emplacement de la source Φ_S, il sera donc possible de calculer le décalage spectral réellement mesuré par l'observateur, différent de celui simplement déduit du rapport des facteurs d'échelle.

3.4 Implications cosmologiques : formulations analytiques et interprétations

3.4.1 Correspondance entre distributions de matière simulée et observée

L'établissement du lien entre les quantités présentes dans une métrique faiblement perturbée et celles calculées dans une simulation cosmologique ainsi que la détermination des équations géodésiques régissant le déplacement de la lumière sous l'effet de la structuration conduisent dès à présent à deux remarques importantes et reliées aux travaux de Chisari et Zaldarriaga[103] et de Bonvin *et al.*[105]. La première remarque concerne d'éventuelles corrections à appliquer sur la structuration de la matière calculée par des simulations newtoniennes et la seconde concerne l'effet du redshift inhomogène sur la distance de luminosité. Autrement dit la première s'intéresse à la façon de corriger la distribution de matière dans les simulations pour obtenir des quantités observables tandis que la seconde s'intéresse à la façon dont ces quantités observables doivent être corrigées pour correctement les interpréter dans un contexte cosmologique.

Comme nous l'avons déjà vu, la différence entre l'équation 3.43 et l'équation de Poisson 3.44 effectivement résolue dans les simulations cosmologiques vient du fait que ces dernières travaillent en jauge comobile et non en jauge newtonienne. Le potentiel Φ, la densité ρ et la vitesse $\vec{v}$ du champ de matière calculés sont exacts[103] au sens d'une métrique faiblement perturbée, hormis qu'une correction $\delta\vec{x}_0$ doit être appliquée à la position $\vec{x}$ des particules lorsqu'il s'agit d'interpréter leur position en jauge newtonienne :

$$\Phi_\mathrm{N} = \Phi_\mathrm{sim} \qquad \vec{v}_\mathrm{N} = \vec{v}_\mathrm{sim} \qquad \vec{x}_\mathrm{N} = \vec{x}_\mathrm{sim} + \delta\vec{x}_0 \tag{3.59}$$

avec :

$$\delta\vec{x}_0 = -\int \frac{\mathrm{d}^3k}{(2\pi)^3}\frac{i\vec{k}}{k^2}\delta_k^0 e^{i\vec{k}\cdot\vec{x}_0} \qquad \text{et} \qquad \delta^0 = -\vec{\nabla}\cdot\delta\vec{x}_0 = -5\Phi_\mathrm{N}^0 \tag{3.60}$$

qui correspond au déplacement initial à appliquer aux particules en jauge newtonienne. En outre, pour corriger les positions des particules (r, θ, φ) et les amener dans le système de coordonnées de l'observateur, il faut, en approximation de Born, appliquer les transformations[103] :

$$\begin{aligned}
r_\mathrm{obs} &= r(\tilde{\eta}) + \delta r = \eta_0 - \tilde{\eta} - 2\int_{\eta_0}^{\tilde{\eta}} \Phi_\mathrm{N}\,\mathrm{d}\eta \\
\theta_\mathrm{obs} &= \theta(\tilde{\eta}) + \delta\theta = \theta(\tilde{\eta}) + 2\int_0^{r(\tilde{\eta})} \mathrm{d}\chi \frac{r(\tilde{\eta}) - \chi}{r(\tilde{\eta})\,\chi}\frac{\partial\Phi_\mathrm{N}}{\partial\theta} \\
\varphi_\mathrm{obs} &= \varphi(\tilde{\eta}) + \delta\varphi = \varphi(\tilde{\eta}) + 2\int_0^{r(\tilde{\eta})} \mathrm{d}\chi \frac{r(\tilde{\eta}) - \chi}{r(\tilde{\eta})\,\chi\sin^2\theta(\tilde{\eta})}\frac{\partial\Phi_\mathrm{N}}{\partial\varphi}
\end{aligned} \tag{3.61}$$

où $\tilde{\eta}$ est le temps conforme « vu » par l'observateur. Ces changements de coordonnées permettent ainsi de passer de la distribution de matière simulée en jauge comobile à la distribution de matière effectivement observée.

3.4.2 Effets sur le redshift et sur la distance de luminosité

Au lieu de chercher à corriger la distribution de matière simulée, il peut être intéressant de chercher à évaluer l'effet de la propagation de la lumière au travers de la distribution de matière sur la mesure du redshift et de la distance de luminosité. C'est l'approche suivie par Bonvin *et al.*[105]. Dans la suite, on se place dans l'approximation de Born, on pose $c = 1$, et l'on note a_S le facteur d'échelle au moment de l'émission de la lumière, a_O le facteur d'échelle au moment de sa réception, $\vec{v}_O$ et $\vec{v}_S$ les champs de vitesse de l'observateur et de la source, $\vec{n}$ la direction de propagation de la lumière et $\mathcal{H} = \frac{a'}{a}$ le

paramètre de Hubble en temps conforme. Dans ces conditions, le redshift mesuré est égal à :

$$z_S = \bar{z}_S + \delta z_S = \frac{a_O}{a_S}\left[\Phi_O - \Phi_S + (\vec{v}\cdot\vec{n})_O - (\vec{v}\cdot\vec{n})_S - 2\int_{\lambda_S}^{\lambda_O}\frac{\partial\Phi}{\partial\eta}\,\mathrm{d}\lambda\right] \tag{3.62}$$

où $\bar{z}_S = \frac{a_O}{a_S} - 1$ est le redshift homogène. Ce redshift dépend donc :

- de la différence des potentiels locaux de l'observateur Φ_O et de la source Φ_S
- de la vitesse relative de l'un par rapport à l'autre projetée sur la ligne de visée
- de l'intégrale de la dérivée du potentiel par rapport au temps conforme le long de la trajectoire

À partir de ce redshift, il est possible de calculer la distance de luminosité en tenant compte des inhomogénéités :

$$\begin{aligned}
d_L(z_S,\vec{n}) =& d_L(\bar{z}_S,\vec{n}) + \frac{\mathrm{d}}{\mathrm{d}\bar{z}_S}d_L(z_S,\vec{n})\,\delta z_S \\
=& (1+z_S)(\eta_O-\eta_S)\left[1 - \frac{1}{(\eta_O-\eta_S)\mathcal{H}_S}\vec{v}_O\cdot\vec{n} - \left(1-\frac{1}{(\eta_O-\eta_S)\mathcal{H}_S}\right)\vec{v}_S\cdot\vec{n}\right. \\
&- \left(1-\frac{1}{(\eta_O-\eta_S)\mathcal{H}_S}\right)\Phi_S - \frac{1}{(\eta_O-\eta_S)\mathcal{H}_S}\Phi_O \\
&+ \frac{2}{\eta_O-\eta_S}\int_{\eta_S}^{\eta_O}\Phi\,\mathrm{d}\eta + \frac{2}{(\eta_O-\eta_S)\mathcal{H}_S}\int_{\eta_S}^{\eta_O}\frac{\partial\Phi}{\partial\eta}\,\mathrm{d}\eta \\
&- 2\int_{\eta_S}^{\eta_O}\frac{\eta-\eta_S}{\eta_O-\eta_S}\frac{\partial\Phi}{\partial\eta}\,\mathrm{d}\eta + \int_{\eta_S}^{\eta_O}\frac{(\eta-\eta_S)(\eta_O-\eta)}{\eta_O-\eta_S}\frac{\partial^2\Phi}{\partial\eta^2}\,\mathrm{d}\eta \\
&\left.- \int_{\eta_S}^{\eta_O}\frac{(\eta-\eta_S)(\eta_O-\eta)}{\eta_O-\eta_S}\nabla^2\Phi\,\mathrm{d}\eta\right]
\end{aligned} \tag{3.63}$$

Cette expression analytique permet de mettre en lumière la signification physique des termes influant sur la distance de luminosité inhomogène :

- les vitesses relatives de l'observateur et de la source sur la première ligne
- la différence des potentiels locaux à l'emplacement de la source et de l'observateur sur la seconde ligne
- l'intégrale de Φ et de ses dérivées temporelles le long de la trajectoire sur les troisième et quatrième lignes
- le lentillage causé par la distribution de matière sur la dernière ligne

Par le biais de tous ces termes, l'identification directe d'une telle distance à une distance de luminosité homogène peut mener à une mauvaise estimation des paramètres cosmologiques. D'autre part, la distribution des distances de luminosité inhomogènes, c'est-à-dire le spectre de puissance de la distance de luminosité, porte en lui-même une information sur la structuration. Il s'agit donc d'une observable pertinente pour sonder la cosmologie et l'accélération de l'expansion. Le prochain chapitre se penchera justement sur tous ces aspects en mettant en œuvre l'application du formalisme présenté dans les pages précédentes au suivi de la lumière dans des simulations cosmologiques.

Chapitre 4

L'Univers observé : empreinte des inhomogénéités sur les mesures de distance

***

4.1 Méthodologie et approximations 70
4.1.1 Principe du backward raytracing en champ faible 70
4.1.2 Obtention des jeux de données 74
4.1.3 Hypothèses et approximations 76
4.2 Résultats 80
4.2.1 Effets sur le redshift 80
4.2.2 Effets sur les distances cosmologiques 81
4.3 Implications cosmologiques et perspectives 86
4.3.1 Bilan 86
4.3.2 Perspectives 86
4.3.3 Conclusion 87

Nous avons vu au chapitre précédent que la structuration de l'Univers entraîne des perturbations de la métrique homogène pour lesquelles il est possible d'obtenir un système d'équations décrivant la trajectoire d'un photon dans l'espace-temps. Intégrer directement cette trajectoire, c'est accéder aux nombreuses conséquences observationnelles résultant de la propagation de la lumière au sein de la structure filamentaire de l'Univers par une approche unifiée. Or, comme cela a pu être montré ne serait-ce qu'au travers de la fonction de masse, différentes cosmologies réalistes au sens de leur compatibilité avec les observations peuvent conduire à des grandes structures aux caractéristiques sensiblement différentes. Dans ce contexte, la cosmologie laisse son empreinte dans ce qui est observé, mais aussi dans la façon même d'observer : la propagation de la lumière est intrinsèquement porteuse d'information cosmologique. Déterminer sa dépendance à la cosmologie permet donc d'une part, de mieux estimer les biais sur la mesure des paramètres avec les sondes actuelles, mais aussi de s'en servir comme d'une nouvelle sonde pour explorer les propriétés de l'accélération de l'expansion. Dans cette optique, ce chapitre est consacré à l'étude des inhomogénéités sur la mesure des distances cosmologiques. Comme souligné à la fin du chapitre précédent, la nature et la dépendance de ces effets a déjà fait l'objet d'études analytiques[105]. Des recherches de solutions exactes ou approchés dans des modèles d'univers inhomogènes en Swiss-Cheese[106] ou en réseau[107] ont également été menées. Si ces travaux permettent de mettre en évidence l'existence et la nature d'effets mesurables, les approximations sur lesquels ils

reposent telles que celle de Born ou celle d'une distribution d'inhomogénéités rigoureusement périodique laissent la question de la quantification de ces effets en suspens. Aussi, comme nous allons le voir, la résolution numérique des équations géodésiques dans des simulations de la taille de l'Univers Observable va permettre d'apporter un éclairage nouveau sur cette question.

Pour traiter de ces aspects, le présent chapitre est composé de trois grandes sections. La première est consacrée à la présentation de la méthode d'analyse utilisée et des approximations qu'elle met en jeu. Il s'agit là de se focaliser sur les problématiques physiques, les aspects numériques et les aspects de validation étant décrits dans la seconde partie du manuscrit. La seconde section présente les résultats obtenus sur la mesure du redshift et la mesure des distances de diamètre angulaire pour les trois modèles cosmologiques du Full Universe Run. Enfin, la troisième partie discute les implications cosmologiques de ces résultats et donne des pistes d'ouverture et d'extension dans la mesure où l'on dispose d'une approche générique pour traiter de la propagation de la lumière dans une distribution arbitraire de potentiel.

4.1 Méthodologie et approximations

4.1.1 Principe du backward raytracing en champ faible

De l'observateur à la source

La première étape vers la réalisation de simulations de propagation de la lumière en champ faible est de parvenir à déterminer la géodésique nulle entre une source S et un observateur O. Pour cela, deux approches équivalentes du fait de la réversibilité des équations et déjà évoquées au chapitre précédent sont possibles :

- une approche de « forward raytracing », c'est-à-dire l'intégration de la trajectoire de façon « naturelle », depuis la source vers l'observateur
- une approche de « backward raytracing », c'est-à-dire l'intégration de la trajectoire à rebours, depuis l'observateur jusqu'à la source

Du fait des perturbations de la métrique, on ne peut savoir *a priori* la trajectoire que va suivre le photon, et c'est d'ailleurs pour cela que l'on cherche à intégrer son mouvement. Cela implique qu'il est impossible de fixer *a priori* un autre point de passage étant donnée une position initiale. Autrement dit soit l'on fixe la source, soit l'on fixe l'observateur. Dans le cas où l'on souhaiterait accéder à la géodésique nulle liant deux points de l'espace-temps, il faudrait procéder à une recherche dichotomique de la courbe liant ces deux points, ce qui impliquerait un grand nombre d'opérations pour chaque trajectoire. Dans notre cas, l'objectif est de connaître les effets des inhomogénéités sur les distances cosmologiques mesurées par un observateur : c'est donc la position de ce dernier qui sera fixée et la trajectoire sera intégrée à rebours. De plus, il ne s'agit pas de mesurer les déformations subies par l'image d'un objet en particulier, surtout que, formellement, aucune source lumineuse n'est présente dans les simulations considérées ici du fait de la non prise en compte des processus thermiques. Il s'agit au contraire de mesurer les déformations dans tout l'espace autour de l'observateur, et ce pour tous les redshifts, en vue d'en extraire des propriétés statistiques indépendantes de la direction de propagation.

Couverture du ciel

Pour se faire, il faut d'une part échantillonner les directions d'observation (θ, φ) autour de l'observateur, et d'autre part échantillonner la profondeur d'observation en redshift z. L'obtention du second échantillonnage est immédiate à partir du moment où l'on intègre une trajectoire puisqu'à l'issue du procédé, on dispose de la valeur de toutes les quantités physiques le long de la géodésique : coordonnées, redshift, facteur d'échelle, potentiel... L'obtention du premier, quant à elle, demande de lancer un grand nombre de rayons dans toutes les directions autour de l'observateur. Pour y parvenir, on choisit

une distribution de points D_p sur la sphère unitaire centrée autour de l'observateur O. Chaque point peut être alors associé à un vecteur :

$$\vec{k}_\emptyset = \left[\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)_\emptyset, \left(\frac{\mathrm{d}x}{\mathrm{d}\lambda}\right)_\emptyset, \left(\frac{\mathrm{d}y}{\mathrm{d}\lambda}\right)_\emptyset, \left(\frac{\mathrm{d}z}{\mathrm{d}\lambda}\right)_\emptyset\right] \quad \text{avec} \quad \begin{cases} \left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)_\emptyset = 1 \\ \sqrt{\left(\frac{\mathrm{d}x}{\mathrm{d}\lambda}\right)_\emptyset^2 + \left(\frac{\mathrm{d}y}{\mathrm{d}\lambda}\right)_\emptyset^2 + \left(\frac{\mathrm{d}z}{\mathrm{d}\lambda}\right)_\emptyset^2} = 1 \end{cases} \tag{4.1}$$

où l'on a fixé la coordonnée temporelle à 1 tandis que les coordonnées spatiales ont été déterminées à partir de la position du point sur la sphère. Le vecteur $\vec{k}_0$ initial est obtenu par renormalisation de $\vec{k}_\emptyset$ de façon à avoir $\frac{\mathrm{d}s^2}{\mathrm{d}\lambda} = 0$. Dans la suite, on pose :

$$\boxed{\Xi = \frac{\left(1 - 2\frac{\Phi}{c^2}\right)\left[\left(\frac{\mathrm{d}x}{\mathrm{d}\lambda}\right)^2 + \left(\frac{\mathrm{d}y}{\mathrm{d}\lambda}\right)^2 + \left(\frac{\mathrm{d}z}{\mathrm{d}\lambda}\right)^2\right]}{c^2\left(1 + 2\frac{\Phi}{c^2}\right)\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2}} \tag{4.2}$$

si bien que les conditions initiales de chaque trajectoire sont données par :

$$\vec{x}_0 = [0, x_O, y_O, z_O] \qquad \text{et} \qquad \vec{k}_0 = \left[\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)_\emptyset, \frac{1}{\Xi_\emptyset}\left(\frac{\mathrm{d}x}{\mathrm{d}\lambda}\right)_\emptyset, \frac{1}{\Xi_\emptyset}\left(\frac{\mathrm{d}y}{\mathrm{d}\lambda}\right)_\emptyset, \frac{1}{\Xi_\emptyset}\left(\frac{\mathrm{d}z}{\mathrm{d}\lambda}\right)_\emptyset\right] \tag{4.3}$$

Dans ce cadre, échantillonner (θ, φ) revient à utiliser une distribution aléatoire de N_p points pour paver uniformément la surface de la sphère unitaire. En dimension n, les coordonnées $[x_1, \ldots, x_n]$ de tels points sur la surface d'une hypersphère peuvent être obtenues à partir de n variables aléatoires gaussiennes $[\tilde{x}_1, \ldots, \tilde{x}_n]$ par[108] :

$$\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \frac{1}{\sqrt{\sum_{i=1}^n \tilde{x}_i^2}} \begin{bmatrix} \tilde{x}_1 \\ \vdots \\ \tilde{x}_n \end{bmatrix} \tag{4.4}$$

Le choix de N_p permet d'ajuster la précision de l'échantillonnage : pour une simulation de N^3 particules associées à un maillage cubique de base de N^3 cellules, un observateur central, et une propagation de photons jusqu'à la surface de la sphère inscrite dans la boîte de simulation, $N_p \sim 4\pi\left(\frac{N}{2}\right)^2$ conduit à une résolution en termes de géodésiques nulles comparable à la résolution spatiale du champ gravitationnel comme illustré en figure 32. Il s'agit là d'une limite haute de résolution : pour les simulations 8192^3 du Full Universe Run, cela correspond à un peu plus de 200 millions de trajectoires. Dans la pratique, prendre N_p de l'ordre du million ou de la dizaine de millions suffit à échantillonner convenablement le ciel et assurer la convergence des résultats présentés dans cette thèse.

Intégration des équations géodésiques

Une fois les photons lancés depuis l'observateur, il s'agit d'intégrer les équations géodésiques par rapport au paramètre affine λ :

$$\begin{bmatrix} \eta \\ x \\ y \\ z \\ \frac{\mathrm{d}\eta}{\mathrm{d}\lambda} \\ \frac{\mathrm{d}x}{\mathrm{d}\lambda} \\ \frac{\mathrm{d}y}{\mathrm{d}\lambda} \\ \frac{\mathrm{d}z}{\mathrm{d}\lambda} \end{bmatrix}_{n+1} = \begin{bmatrix} \eta \\ x \\ y \\ z \\ \frac{\mathrm{d}\eta}{\mathrm{d}\lambda} \\ \frac{\mathrm{d}x}{\mathrm{d}\lambda} \\ \frac{\mathrm{d}y}{\mathrm{d}\lambda} \\ \frac{\mathrm{d}z}{\mathrm{d}\lambda} \end{bmatrix}_n + f\left(\begin{bmatrix} \eta \\ x \\ y \\ z \\ \frac{\mathrm{d}\eta}{\mathrm{d}\lambda} \\ \frac{\mathrm{d}x}{\mathrm{d}\lambda} \\ \frac{\mathrm{d}y}{\mathrm{d}\lambda} \\ \frac{\mathrm{d}z}{\mathrm{d}\lambda} \end{bmatrix}_n\right)\mathrm{d}\lambda \qquad \text{ou} \qquad \begin{bmatrix} a \\ \eta \\ x \\ y \\ z \\ \frac{\mathrm{d}\eta}{\mathrm{d}\lambda} \\ \frac{\mathrm{d}x}{\mathrm{d}\lambda} \\ \frac{\mathrm{d}y}{\mathrm{d}\lambda} \\ \frac{\mathrm{d}z}{\mathrm{d}\lambda} \end{bmatrix}_{n+1} = \begin{bmatrix} a \\ \eta \\ x \\ y \\ z \\ \frac{\mathrm{d}\eta}{\mathrm{d}\lambda} \\ \frac{\mathrm{d}x}{\mathrm{d}\lambda} \\ \frac{\mathrm{d}y}{\mathrm{d}\lambda} \\ \frac{\mathrm{d}z}{\mathrm{d}\lambda} \end{bmatrix}_n + f\left(\begin{bmatrix} a \\ \eta \\ x \\ y \\ z \\ \frac{\mathrm{d}\eta}{\mathrm{d}\lambda} \\ \frac{\mathrm{d}x}{\mathrm{d}\lambda} \\ \frac{\mathrm{d}y}{\mathrm{d}\lambda} \\ \frac{\mathrm{d}z}{\mathrm{d}\lambda} \end{bmatrix}_n\right)\mathrm{d}\lambda \tag{4.5}$$

suivant que l'on souhaite, ou non, calculer l'évolution du facteur d'échelle conjointement avec le même schéma numérique. En réalité, on utilisera plutôt un schéma d'intégration de Runge-Kutta d'ordre 4, mais le principe reste identique. La forme de f correspond aux expressions des dérivées secondes des coordonnées données au chapitre précédent. Dans le cas des géodésiques nulles, on a ainsi l'expression suivante où a et sa dérivée sont donnés par le modèle d'évolution homogène étudié, où les dérivées partielles de Φ sont issues des valeurs provenant de la grille de simulation et où la dérivée totale de Φ est calculée par la variation du potentiel d'un pas d'intégration à l'autre :

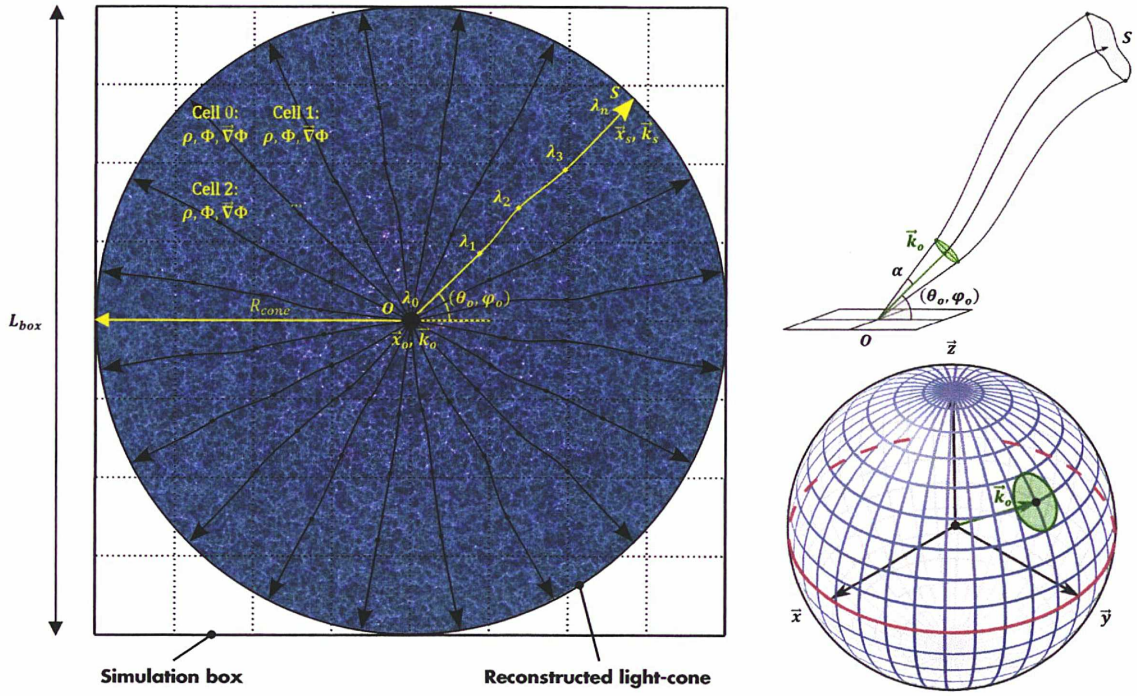
$$f\left(\begin{bmatrix}\eta\\ x\\ y\\ z\\ \frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\\ \frac{\mathrm{d}x}{\mathrm{d}\lambda}\\ \frac{\mathrm{d}y}{\mathrm{d}\lambda}\\ \frac{\mathrm{d}z}{\mathrm{d}\lambda}\end{bmatrix}\right) = \begin{bmatrix} \frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\\ \frac{\mathrm{d}x}{\mathrm{d}\lambda}\\ \frac{\mathrm{d}y}{\mathrm{d}\lambda}\\ \frac{\mathrm{d}z}{\mathrm{d}\lambda}\\ -\frac{2a'(\eta)}{a(\eta)}\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2 - \frac{2}{c^2}\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\left(\frac{\partial\Phi(\eta,x,y,z)}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}\lambda} + \frac{\partial\Phi(\eta,x,y,z)}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}\lambda} + \frac{\partial\Phi(\eta,x,y,z)}{\partial z}\frac{\mathrm{d}z}{\mathrm{d}\lambda}\right)\\ -\frac{2a'(\eta)}{a(\eta)}\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\frac{\mathrm{d}x}{\mathrm{d}\lambda} + \frac{2}{c^2}\frac{\mathrm{d}\Phi(\lambda)}{\mathrm{d}\lambda}\frac{\mathrm{d}x}{\mathrm{d}\lambda} - 2\frac{\partial\Phi(\eta,x,y,z)}{\partial x}\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2\\ -\frac{2a'(\eta)}{a(\eta)}\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\frac{\mathrm{d}y}{\mathrm{d}\lambda} + \frac{2}{c^2}\frac{\mathrm{d}\Phi(\lambda)}{\mathrm{d}\lambda}\frac{\mathrm{d}y}{\mathrm{d}\lambda} - 2\frac{\partial\Phi(\eta,x,y,z)}{\partial y}\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2\\ -\frac{2a'(\eta)}{a(\eta)}\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\frac{\mathrm{d}z}{\mathrm{d}\lambda} + \frac{2}{c^2}\frac{\mathrm{d}\Phi(\lambda)}{\mathrm{d}\lambda}\frac{\mathrm{d}z}{\mathrm{d}\lambda} - 2\frac{\partial\Phi(\eta,x,y,z)}{\partial z}\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2 \end{bmatrix} \tag{4.6}$$

Pour plus de généricité, il est bien entendu possible d'utiliser le système valide pour toute valeur de $\mathrm{d}s^2$. Étant donné que la grille est fixe et n'est pas modifiée au cours du calcul, il est possible de s'en servir pour déterminer le pas $\mathrm{d}\lambda$ de façon à avoir environ N_λ pas d'intégration par cellule. Cela permet, dans le cadre d'une grille adaptative raffinée sur un critère de densité, de maximiser la précision du calcul dans les zones structurées. De plus, les valeurs de Φ et de ses dérivées ne sont pas simplement lues dans la grille : elles sont réinterpolées en vue de garantir la continuité du potentiel. Cela assure également la cohérence numérique avec les schémas d'interpolation utilisés dans le code dynamique. Dans la suite on prendra $N_\lambda = 4$ ou $N_\lambda = 8$, conformément aux tests présentés au dernier chapitre, et on utilisera une interpolation linéaire tridimensionnelle, c'est-à-dire un « Could-in-Cell » inverse[109]. Le principe de l'intégration est résumé visuellement sur la partie gauche de la figure 32.

Propagation d'un faisceau de lumière

Obtenir des trajectoires de photons dans toutes les directions permet de procéder à un certain nombre d'analyses, mais des quantités comme la distance de diamètre angulaire et la distance de luminosité nécessitent des éléments supplémentaires. En effet, la détermination de ces deux grandeurs passe par le calcul de déformation de faisceaux infiniment fins. Dans un premier temps, l'on pourrait simplement regrouper les trajectoires obtenues précédemment en faisceaux de lumière. Le problème est qu'une dizaine de millions de trajectoires ne suffit pas à échantillonner le ciel à la résolution nécessaire. Pour le résoudre, on fait appel à une approche « ray bundle »[110] : pour chaque trajectoire intégrée, on calcule la propagation de N_b photons, avec $N_b = 8$ dans la suite, répartis uniformément sur un cercle centré initialement sur cette trajectoire comme illustré sur la partie droite de la figure 32. Les géodésiques sont intégrées de façon indépendante par la méthode exposée dans le paragraphe précédent, mais cette fois, il est possible d'avoir accès aux déformations subies par le cercle initial le long de la géodésique. À partir de ces informations, on peut calculer la variation de la taille du faisceau, mais aussi éventuellement la convergence et le cisaillement dans le formalisme du lentillage. Pour déterminer la distance de diamètre angulaire, seule la première quantité suffit : on la calcule comme la moyenne des distances entre le photon central et les N_b photons périphériques.

Le faisceau est paramétré par l'angle d'ouverture à l'origine α annoté en figure 32. Les vecteurs



(a) Trajectoire d'un photon dans un cône de lumière.

(b) Lancement d'un faisceau de rayons.

FIGURE 32 – Principe du « backward raytracing » et de l'approche « ray bundle ». Pour sonder l'effet des inhomogénéités sur la propagation de la lumière, on lance N_p photons de façon quasi-uniforme autour de l'observateur de façon à échantillonner les $\sim 40\,000$ degrés carrés de la sphère céleste. Pour une simulation de N^3 cellules, prendre $N_p \sim 4\pi \left(\frac{N}{2}\right)^2$ permet de s'assurer d'avoir une trajectoire dans chaque cellule intersectant la surface du cône, comme ci-dessus, à gauche. Chacun des photons est propagé à rebours de l'observateur vers la source par intégration des équations géodésiques à partir des conditions initiales $\left(\vec{x}_0, \vec{k}_0\right)$. La valeur du potentiel le long de la trajectoire est réinterpolée à partir des données extraites des maillages issus des simulations cosmologiques. L'échantillonnage en redshift provient naturellement du calcul de la trajectoire car à chaque pas d'intégration λ_i, toutes les quantités physiques d'intérêt sont extraites ou calculées : $\vec{x}_i$, $\vec{k}_i$, z_i, a_i, ρ_i, Φ_i, $\vec{\nabla}\Phi_i$... Ainsi, l'intégration ne s'arrête pas lors de la rencontre éventuelle d'une source : elle continue jusqu'à une certaine limite de redshift z_{max} ce qui permet d'obtenir les données nécessaires pour toutes les sources distribuées le long de cette trajectoire, de z_O à z_{max}. Cependant, des quantités comme la distance de diamètre angulaire, et donc aussi la distance de luminosité, ne reposent pas sur des trajectoires uniques, mais au contraire sur les déformations subies par des faisceaux de lumière infiniment fins. Dans ce cadre, une approche de « ray bundle » est utilisée, comme illustré à droite : un cercle de N_b photons est lancé autour d'un photon central comme représenté sur la sphère. Dans la pratique, on prendra $N_b = 8$. La taille du faisceau est paramétré par l'angle α annoté sur le schéma du dessus. La trajectoire de chaque photon est intégrée indépendamment puis, à l'issue de ce premier calcul, une seconde passe permet de déterminer les déformations subies par le cercle original pour chaque redshift de la trajectoire centrale.

DATA EXTRACTION PARAMETERS IN THE FULL UNIVERSE RUN					
	x_O	y_O	z_O	a_O	a_{lim}
Cone 1	$0.5L_{\text{box}}$	$0.5L_{\text{box}}$	$0.5L_{\text{box}}$	1.00	0.032
Cone 2	$0.5L_{\text{box}}$	$0.1L_{\text{box}}$	$0.1L_{\text{box}}$	1.00	0.032
Cone 3	$0.1L_{\text{box}}$	$0.5L_{\text{box}}$	$0.9L_{\text{box}}$	1.00	0.032
Cone 5	$0.5L_{\text{box}}$	$0.5L_{\text{box}}$	$0.5L_{\text{box}}$	0.66	0.032
Cone 6	$0.5L_{\text{box}}$	$0.5L_{\text{box}}$	$0.9L_{\text{box}}$	0.45	0.032
	x_c	y_c	z_c	a	L
Sample	$0.5L_{\text{box}}$	$0.5L_{\text{box}}$	$0.5L_{\text{box}}$	all	$0.125L_{\text{box}}$

FIGURE 33 – Paramètres d'extraction des cônes et du « sample » issus des simulations. Pour chaque modèle cosmologique, cinq cônes ont été produits, pour des observateurs à différentes positions dans l'espace et dans le temps. En plus, les données d'un sous-volume central ont été extraites à tous les pas de temps.

directeurs initiaux des photons le constituant sont quant à eux donnés par :

$$\begin{aligned}
\vec{k}_{\emptyset_i} \cdot \vec{e}_x &= (-\cos\theta\cos\phi\sin\alpha\sin\beta - \sin\phi\sin\alpha\cos\beta)\cos\delta_i \\
&\quad + (-\cos\theta\cos\phi\sin\alpha\cos\beta + \sin\phi\sin\alpha\sin\beta)\sin\delta_i + \sin\theta\cos\phi\cos\alpha \\
\vec{k}_{\emptyset_i} \cdot \vec{e}_y &= (-\cos\theta\sin\phi\sin\alpha\sin\beta + \cos\phi\sin\alpha\cos\beta)\cos\delta_i \\
&\quad + (-\cos\theta\sin\phi\sin\alpha\cos\beta - \cos\phi\sin\alpha\sin\beta)\sin\delta_i + \sin\theta\sin\phi\cos\alpha \\
\vec{k}_{\emptyset_i} \cdot \vec{e}_z &= \sin\theta\sin\alpha\sin\beta\cos\delta_i + \sin\theta\sin\alpha\cos\beta\sin\delta_i + \cos\theta\cos\alpha
\end{aligned} \tag{4.7}$$

avec :

$$\theta = \operatorname{acos}\left(\vec{k}_\emptyset \cdot \vec{e}_z\right) \qquad \phi = \operatorname{atan2}\left(\vec{k}_\emptyset \cdot \vec{e}_y, \vec{k}_\emptyset \cdot \vec{e}_x\right) \qquad \delta_i = \frac{2i\pi}{N_b} \tag{4.8}$$

où $\vec{k}_\emptyset$ est le vecteur directeur du photon central donné en 4.1, où $i \in [0, N_b[\cap \mathbb{N}$ est l'index du photon courant et où β spécifie l'emplacement des photons sur le cercle, choisi aléatoirement pour chaque faisceau dans la suite en vue d'éviter toute direction préférentielle.

À l'issue de cette procédure il est donc possible de connaître les déformations subies par des faisceaux lumineux dans toutes les directions de la voûte céleste et pour tous les redshifts, ce qui donne, de fait, accès à la mesure des distances cosmologiques dans tout l'espace étant donnée une distribution arbitraire de potentiel.

4.1.2 Obtention des jeux de données

Dépasser les approximations usuelles

Afin de pouvoir comparer l'effet de différents niveaux de structuration sur la propagation de la lumière, nous allons utiliser les données de simulation issues du Full Universe Run dont il a déjà été question précédemment. L'objectif est également, d'éviter, autant que faire se peut, les approximations usuelles décrites au chapitre précédent. D'une part, grâce à la taille des boîtes de simulation, $L_{\text{box}} = 21\,\text{Gpc/h}$, il va être possible de construire des cônes de lumière sans aucune duplication de données, contrairement à ce qui avait pu être fait jusqu'à maintenant. Dans la pratique on prendra $z_{max} \approx 30$, les premières sources observables se formant de toute façon vers $z \approx 10$ et quand bien même nous aurions besoin du champ entre $z = 30$ et $z = z_{\text{CMB}}$ celui-ci pourrait être reconstruit dans l'approximation de Zel'dovich. D'autre part, au lieu de reconstruire le potentiel à partir de la distribution de matière en post-traitement, nous allons directement extraire la valeur du champ calculé par le code dynamique. Ce faisant, on maximise la précision du calcul du fait que le code utilisé implémente un maillage adaptatif qui permet de suivre l'effondrement de la matière. Les interactions gravitationnelles sont ainsi prises en compte depuis l'échelle de la boîte jusqu'aux niveaux les plus raffinés. En couplant cette extraction à l'intégration décrite dans la section précédente, on s'affranchit donc de la « multi-lens plane approximation » qui avait été mise en œuvre jusqu'à présent dans la très grande majorité des études cherchant à aller au-delà de l'approximation de Born. En d'autres termes, il s'agit d'extraire la distribution du champ gravitationnel et de ses dérivées à la résolution maximale, notamment dans les régions surdenses, pour ensuite y intégrer directement les équations géodésiques sous leur forme la plus générale pour une métrique de champ faible développée au premier ordre.

De l'espace comobile à l'espace des redshifts

Toutefois, une question reste en suspens : comment obtenir les données dans l'espace des redshifts alors que les simulations sont effectuées dans l'espace comobile ? Une première solution imaginable serait d'intégrer les géodésiques au cours de la simulation : sauf que, d'une part, les calculateurs actuels ne disposent pas de la mémoire nécessaire pour effectuer le calcul dynamique et le suivi de la lumière conjointement pour une simulation de la taille de l'Univers Observable, et que d'autre part, cette solution n'est pas compatible avec du backward raytracing. En effet, dans le cas d'une simulation

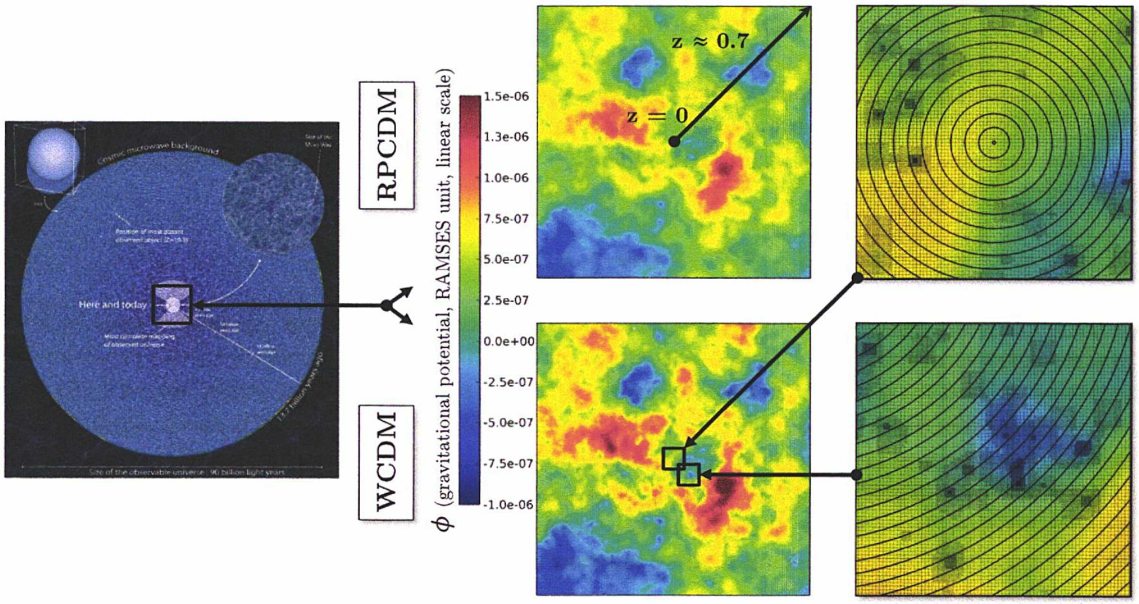


Figure 34 – Potentiel gravitationnel obtenu après construction des cônes de lumière. En appliquant la méthode d'extraction présentée en figure 35, la valeur du potentiel gravitationnel est obtenue dans tout l'espace des redshifts. Celle-ci est distribuée sur une grille de maillage adaptatif d'autant plus raffinée que la densité de matière est élevée, ce qui conduit à disposer d'une plus grande résolution spatiale dans les zones de structuration. Une telle grille est représentée sur la partie droite de la figure. Les cercles concentriques correspondent au raccordement des coquilles. On constate qu'au centre du cône l'épaisseur des coquilles n'est que de trois ou quatre cellules non raffinées : l'échantillonnage en redshift à proximité de l'observateur, c'est-à-dire dans les zones les plus structurées est donc de très bonne qualité comparativement à ce qui aurait pu être obtenu à partir des sauvegardes de tout l'espace comobile des simulations. C'est au sein de cette grille que le suivi de la propagation de la lumière va être effectué, avec plusieurs pas $\mathrm{d}\lambda$ par cellule, y compris dans les niveaux raffinés pour bénéficier d'une résolution plus élevée dans les zones correspondantes. La comparaison des deux figures centrales permet d'évaluer visuellement les différences de potentiel entre le modèle de quintessence de Ratra-Peebles et le modèle phantom. On constate des écarts de potentiel moins élevés dans le premier que dans le second, ce qui est compatible avec les niveaux de structuration présentés dans les chapitres précédents.

dynamique, la situation de l'observateur à $z = 0$ n'est connue qu'à la fin, alors qu'il s'agit d'une condition initiale pour l'intégrateur des géodésiques. Une seconde solution serait de sauvegarder toute la boîte de simulation à tous les pas de temps. Si cette façon de procéder fonctionnerait effectivement sur le papier, elle est actuellement impossible à mettre en place techniquement. En effet, chaque simulation du Full Universe Run est effectuée en $\sim$ 500 pas de temps principaux [1] : sauvegarder ρ, Φ et $\vec{\nabla}\Phi$ au niveau principal de la grille, c'est-à-dire sans même prendre en compte le maillage adaptatif, sur 500 pas de temps demanderait de l'ordre de 8 Po d'espace de stockage. Dans le cas du Full Universe Run, une telle extraction de données à tous les pas de temps et à tous les niveaux de raffinement a été mise en place sur un sous-volume central de $V_{\text{sample}} = \frac{1}{8^3} = \frac{V_{\text{box}}}{512}$ comme précisé en figure 33, mais faire l'équivalent à l'échelle de toute la boîte est incompatible avec les architectures actuelles : chaque simulation n'a été sauvegardée en totalité qu'à 31 pas de temps. Une troisième possibilité serait d'utiliser ces 31 pas de temps pour retracer l'évolution du potentiel le long de la géodésique, en utilisant des données issues de boîtes de plus en plus proches de l'homogénéité à mesure que l'on s'éloigne de l'observateur, à l'image de la construction de cônes de lumière mise en place par Hilbert *et al.*[99]. Toutefois, cette méthode ne permettrait qu'un échantillonnage grossier en redshift.

Pour aller au-delà de ces propositions et disposer à la fois de la meilleure résolution spatiale, d'un bon échantillonnage en redshift et d'une quantité de données gérable par les architectures actuelles,

1. Dans la pratique, il y a un sous-cyclage des pas de temps en fonction de la structuration : les $\sim$ 500 pas correspondent à des instants où l'ensemble de la simulation est synchronisé.

nous allons extraire les quantités physiques nécessaires au cours des simulations, par couches concentriques successives[1, 2]. En effet, si l'on fixe la position des futurs observateurs avant la simulation, il devient possible, pour chaque pas de temps, de sauvegarder approximativement la portion d'espace comobile intersectant le cône de lumière de cet observateur. Pour un pas de temps amenant la simulation d'un facteur d'échelle a_{old} à un facteur d'échelle a_{new}, cette portion correspond à la coquille sphérique située entre χ_{old} et χ_{new} :

$$\int_{\eta_{\text{new}}}^{\eta_O} c\,\mathrm{d}\eta' = \chi_{\text{new}} < \chi \leq \chi_{\text{old}} = \int_{\eta_{\text{old}}}^{\eta_O} c\,\mathrm{d}\eta' \tag{4.9}$$

où η_O est le temps conforme de l'observateur. À la fin du calcul dynamique on obtient donc ~ 500 couches concentriques pour chaque observateur, qu'il suffit ensuite de fusionner pour produire les cônes de lumières qui serviront de base à la propagation des rayons lumineux, comme décrit en figures 35 et 34. Disposant d'une part d'un intégrateur d'équations géodésiques capable de suivre des faisceaux de lumière et d'assurer la couverture de l'ensemble de la sphère céleste et d'autre part de la distribution du potentiel sur une grille adaptative dans l'espace des redshifts pour plusieurs observateurs virtuels et pour plusieurs cosmologies, nous pouvons procéder à l'analyse de l'empreinte de la structuration sur la mesure des distances avec un nombre minimal d'approximations.

4.1.3 Hypothèses et approximations

Développement au premier ordre d'une métrique faiblement perturbée

Aller au-delà des approximations usuelles ne signifie pas pour autant supprimer toutes les approximations. En effet, un certain nombre de simplifications par rapport à ce que serait la résolution idéale des géodésiques nulles pour une métrique arbitraire demeurent. La première hypothèse importante est celle qui est à la racine du formalisme du chapitre précédent : la prise en compte de faibles perturbations scalaires à un fond de métrique homogène développé au premier ordre. Cette hypothèse permet d'assurer une cohérence vis-à-vis d'une part, du formalisme du lentillage faible et d'autre part vis-à-vis des simulations cosmologiques sous leur forme actuelle. S'attaquer à cette hypothèse, ce qui est tout à fait possible, ce n'est donc pas seulement remettre en cause le formalisme utilisé pour la propagation de la lumière, mais c'est aussi remettre en cause la validité des simulations cosmologiques : dans ce cas, ce serait une toute autre approche qu'il conviendrait de développer, ce qui sort largement du cadre de cette thèse. Le fait de ne développer la métrique qu'au premier ordre permet aussi d'assurer cette cohérence. D'un point de vue purement pratique, il serait aisé d'appliquer la même procédure que celle décrite ici en prenant en compte les termes additionnels du second ordre, et même des ordres supérieurs. Toutefois pour de telles équations, les problématiques liées au choix de jauge peuvent devenir beaucoup plus complexes. De plus, si des termes en $\Phi\nabla^2\Phi$ s'avéraient être non négligeables, cela viendrait remettre en cause, non pas le formalisme utilisé pour la propagation de la lumière, mais les données provenant des simulations. Notons néanmoins que la démarche présentée ici pourrait être reprise pour explorer le cas de métriques plus complexes mais pour des distributions spatiales plus simples, comme par exemple des surdensités gaussiennes distribuées sur des réseaux[107]. Enfin, bien que les calculs aient été effectués avec Ψ et Φ, nous avons finalement posé $\Psi = \Phi$: il s'agit là encore d'assurer la cohérence avec les simulations, bien que le schéma numérique soit en mesure d'intégrer les équations avec deux valeurs distinctes.

Calcul du redshift

Conformément à l'approche de la métrique en champ faible développée au premier ordre et aux calculs du chapitre précédent, le redshift inhomogène est donné par :

$$z = \frac{a_S^2 k_S^0 \left(1 + \frac{\Phi_S}{c^2} - \frac{k_S^i u_S^i}{c k_S^0}\right)}{a_O^2 k_O^0 \left(1 + \frac{\Phi_O}{c^2} - \frac{k_O^i u_O^i}{c k_O^0}\right)} - 1 \tag{4.10}$$

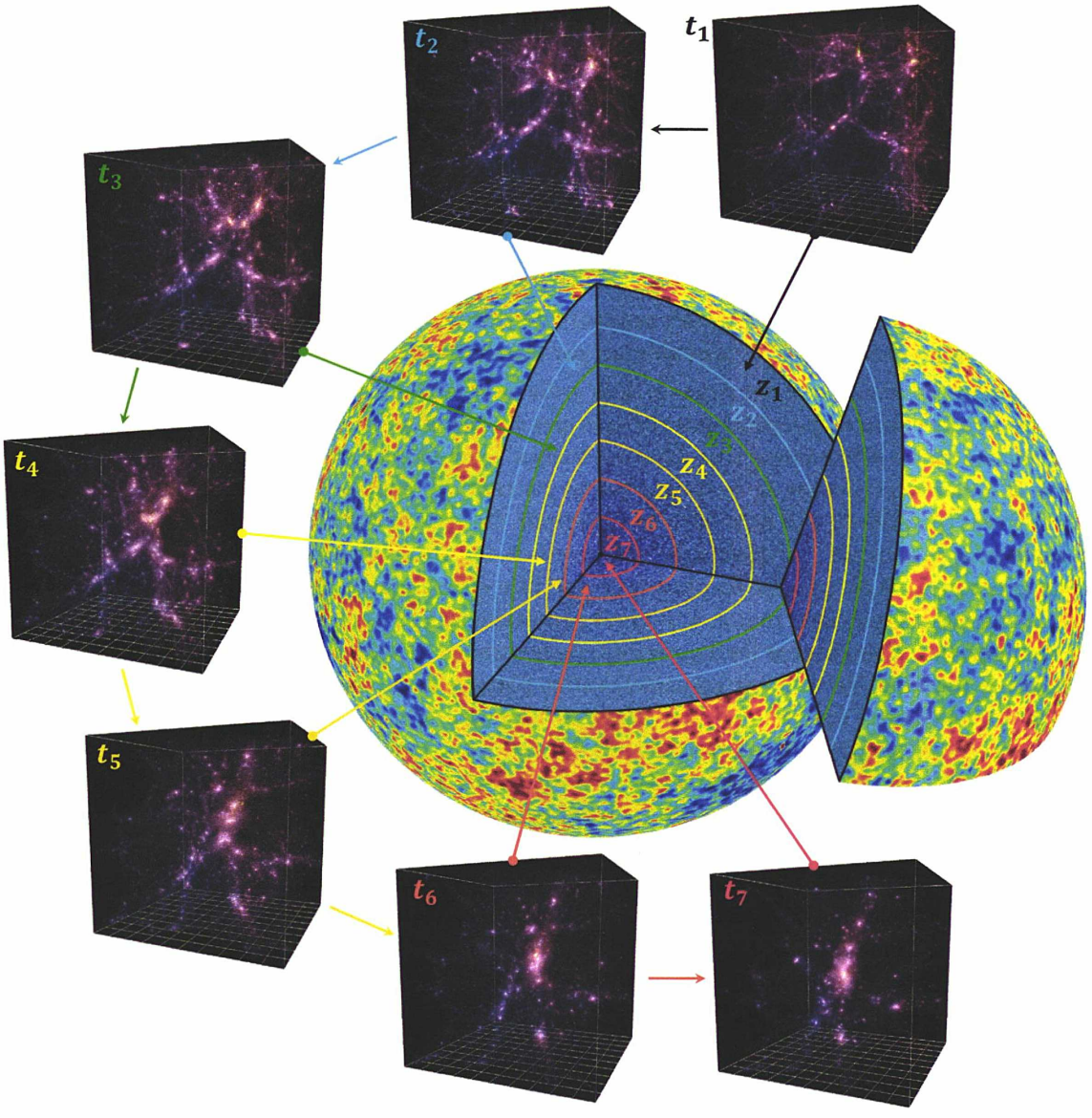


FIGURE 35 – Principe de construction de l'espace des redshifts à partir des boîtes de simulation comobiles. Pour bénéficier d'une grande résolution en redshifts, la position des observateurs virtuels est fixée avant la simulation. Pour chacun d'entre eux, on extrait à chaque pas de temps une coquille de données au sein de la boîte. Ainsi, au début de la simulation, ce sont des coquilles lointaines, non structurées qui sont extraites, puis à mesure que le temps cosmique avance, des portions de plus en plus proches et de plus en plus structurées sont sauvegardées. Pour les simulations utilisées dans la suite, le nombre total de coquilles est de l'ordre de ~ 500 . À la fin du calcul dynamique, les coquilles sont fusionnées de manière à reconstituer l'ensemble de la sphère. Le fait de disposer de boîtes de 21 Gpc/h de côté dans le cadre du Full Universe Run permet d'éviter tout duplication de données : les sphères entières tiennent dans les cubes de simulation.

Si l'on tiendra bien compte des potentiels locaux de la source et de l'observateur, on négligera en revanche les mouvements relatifs de ces deux derniers par rapport à l'expansion. Cela nécessiterait des analyses de champs de vitesse des grandes structures qui dépassent le propos de ce manuscrit. La prise en compte de ces vitesses péculières pourrait cependant naturellement faire l'objet d'un approfondissement, mais ici nous le considèreront comme un effet supplémentaire et indépendant sur le redshift. Aussi, le redshift sera calculé dans les simulations par :

$$\boxed{z \approx \frac{a_S^2 k_S^0 \left(1 + \frac{\Phi_S}{c^2}\right)}{a_O^2 k_O^0 \left(1 + \frac{\Phi_O}{c^2}\right)} - 1} \tag{4.11}$$

Approximations liées à la construction des cônes

Dans l'équation 4.6, tous les termes peuvent être calculés : $\vec{x}_{n-1}$ et $\vec{k}_{n-1}$ sont donnés par l'étape d'intégration précédente, a et a' sont obtenus par l'évolution de la cosmologie homogène, $\frac{\partial \Phi}{\partial x}$, $\frac{\partial \Phi}{\partial y}$ et $\frac{\partial \Phi}{\partial z}$ sont réinterpolés à partir des données de la grille de simulation et $\frac{\mathrm{d}\Phi}{\mathrm{d}\lambda}$ peut être calculé en utilisant la différence de valeur du potentiel entre deux étapes d'intégration successives. Il n'y a donc à première vue, aucune raison d'être amené à négliger des termes. La situation est en réalité légèrement plus complexe et notamment à cause de ce dernier terme. En effet, $\frac{\mathrm{d}\Phi}{\mathrm{d}\lambda}$ peut se décomposer comme la somme des dérivées spatiales de Φ et de sa dérivée temporelle. Or, cette dérivée temporelle n'a pas été prise en compte pour le moment : sa valeur a été négligée face aux autres termes. Physiquement, la variation dans le temps de Φ au passage du photon devrait être effectivement significativement moins élevée que les variations spatiales. Toutefois, il serait nécessaire de s'en assurer et cela devra faire l'objet de développements ultérieurs car les données qui permettraient de calculer ces termes sont absentes des espaces de redshifts. Sur ce point, une amélioration possible de la construction présentée en figure 35 consisterait à élargir l'épaisseur des coquilles pour disposer de zones de recouvrement. Ainsi, chaque redshift z_n serait associé à plusieurs valeurs de Φ à des pas de temps successifs. Cela permettrait d'une part de prendre en compte la valeur de $\frac{\partial \Phi}{\partial \eta}$ lors de l'intégration et d'autre part de mener des analyses liées à des effets comme le redshift drift. De telles zones de recouvrement augmenteraient la taille des sauvegardes, mais dans des proportions tout à fait gérables pour les machines actuelles. Mais même sans passer par cette nouvelle façon de construire les espaces des redshifts, le Full Universe Run permettrait de tester ces effets, du moins à bas z. En effet, si nous ne disposons pas de la variation du potentiel dans les cônes, nous en disposons en revanche dans le sous-volume qui a été extrait au centre de la boîte. Ce dernier a été sauvegardé à tous les pas de temps de la simulation : cela signifie que nous pouvons reconstruire des cônes en post-traitement de la manière dont on le souhaite, et notamment en prenant en compte les termes $\frac{\partial \Phi}{\partial \eta}$. Le volume disponible est limité à $\frac{L_{\text{box}}}{8}$ de côté, ce qui correspond à un espace des redshifts allant jusqu'à $z \approx 0.5$, mais ce qui est tout de même suffisant pour tester la prise en compte de termes additionnels dans la zone la plus structurée autour de l'observateur.

Ce sous-volume permettra, à terme, de dépasser une autre approximation : la non-interpolation temporelle du potentiel. En effet, en extrayant le potentiel dans des coquilles sphériques à la manière de ce qui est présenté en figure 35, nous supposons implicitement que les déviations des rayons lumineux sont faibles. Dans le cas de très fortes déviations, les photons pourraient être suffisamment « retardés » pour que le potentiel extrait soit différent du potentiel effectivement subit. Car les surfaces iso-η ne sont pas des sphères : ce sont des sphères légèrement déformées. Dans le cas où ces perturbations n'entraînent pas de différence de plus d'une cellule avec le cas de sphères parfaites iso-r, alors l'approximation tient. Mais si l'on peut raisonnablement penser que ces effets sont négligeables, ils nécessiteraient néanmoins d'être testés, ne serait-ce que pour connaître la valeur exacte de leur amplitude. Comme dit plus haut, de tels tests seraient réalisables avec le « sample » car celui-ci rend possible une réinterpolation temporelle du potentiel en plus de la réinterpolation spatiale actuelle dans la grille. En fait, il faut voir le sous-volume extrait à tous les pas de temps comme un hypercube disposant d'un

maillage adaptatif 4-dimensionnel. La construction de l'espace des redshifts revient à présélectionner les zones d'intérêt dans ce cube en supposant que la lumière est peu déviée. Mais si l'on dispose de toutes les données, comme c'est le cas au centre de la simulation, alors l'intégration des géodésiques peut être directement effectuée au sein de la grille 4D, ce qui pourrait donner accès à des mesures inédites.

Faisceaux de lumière et condition d'arrêt

Deux derniers points essentiels pour le calcul des distances cosmologiques concernent les faisceaux de lumière. La distance de diamètre angulaire est définie comme le rapport d'une distance, le diamètre physique de l'objet, sur un angle, celui sous lequel cet objet est vu par l'observateur :

$$\boxed{d_A = \frac{x}{\theta}} \tag{4.12}$$

Le seul angle qui compte est donc l'angle entre les photons à l'arrivée sur l'observateur, ou, dans notre cas, l'angle d'ouverture initial. La longueur x correspond, dans notre cas, au rayon du faisceau, ou plutôt à la moyenne des distances entre les photons car, du fait du cisaillement, le faisceau originellement circulaire se déforme peu à peu en une ellipse. À ce sujet, une définition alternative de la distance de diamètre angulaire pourrait être formulée en termes d'aire du faisceau A et d'angle solide Ω. Dans la pratique, à moins de déformations très prononcées, auquel cas il faudrait de toute façon diminuer l'angle d'ouverture, cela n'entraîne pas de différence sur les mesures. Numériquement, ces mesures dépendent de deux paramètres : l'angle d'ouverture 2α et le nombre de photons utilisés. Le premier peut être fixé par des études en convergence : s'il est choisi trop grand, les photons peuvent subir des potentiels très différents et le faisceau ne répond plus au critère de finesse nécessaire, s'il est choisi trop petit, alors tous les photons traversent exactement les mêmes cellules ce qui peut rendre les mesures difficiles. Entre ces deux extrêmes, il existe un régime où la dépendance des résultats à α est minimale donnant accès à une mesure précise des distances cosmologiques : dans la suite on se placera bien entendu dans ce régime. Le second, le nombre de photons N_b dans le « bundle » peut lui aussi être rendu optimal par une analyse en convergence. La borne minimale dans le cas où l'on voudrait effectuer une analyse dans le formalisme du lentillage est donnée par le nombre de points nécessaire pour contraindre l'équation d'une ellipse : $N_b \geq 5$. Au-delà, les cœfficients de l'équation peuvent être trouvés par la méthode des moindres carrés. Dans toute la suite, on considère des faisceaux de 8 photons en plus de la trajectoire centrale. Il peut être également utile de noter que, dans notre cas, on mesure la déformation entre un cercle au niveau de l'observateur et une ellipse au niveau de la source : une inversion de la matrice de déformation peut donc éventuellement être nécessaire si l'on souhaite faire le lien avec les études de lentillage gravitationnel.

Le second et dernier point, plus subtil, et qui nécessiterait une étude à part entière, est le critère d'arrêt du faisceau, dont le choix n'est pas contraint et qui a été relativement peu documenté dans la littérature. La question est la suivante : que considère-t-on comme une tranche de faisceau ? Ou dit autrement, doit-on couper le faisceau par des plans orthogonaux à $\vec{k}$ ou par des isosurfaces de λ, de η, de χ, de a ou de z ? Dans le cas homogène, la question ne se pose par car tous ces critères sont équivalent. Mais dans le cas inhomogène avec des faisceaux dotés d'un angle d'ouverture initial 2α, la question se pose. Arrêter tous les photons du faisceau sur un critère de distance comobile ou de facteur d'échelle homogène alors que la propagation de la lumière est traitée dans le cas inhomogène aurait peu de sens. De même, couper le faisceau par un plan orthogonal aux composantes spatiales de $\vec{k}$, c'est considérer que le potentiel local est nul. Quant à η, il n'est pas mesurable. Reste donc le redshift z qui est directement observable, qui permet de tenir compte des perturbations du potentiel local et qui a un sens physique lorsque toutes les quantités cosmologiques sont exprimées en fonction de ce dernier : c'est donc ce critère d'arrêt qui a été retenu. Il est intéressant de noter qu'observationnellement, ce critère d'arrêt dépend également de la source : il n'est pas le même pour des supernovae, pour les oscillations

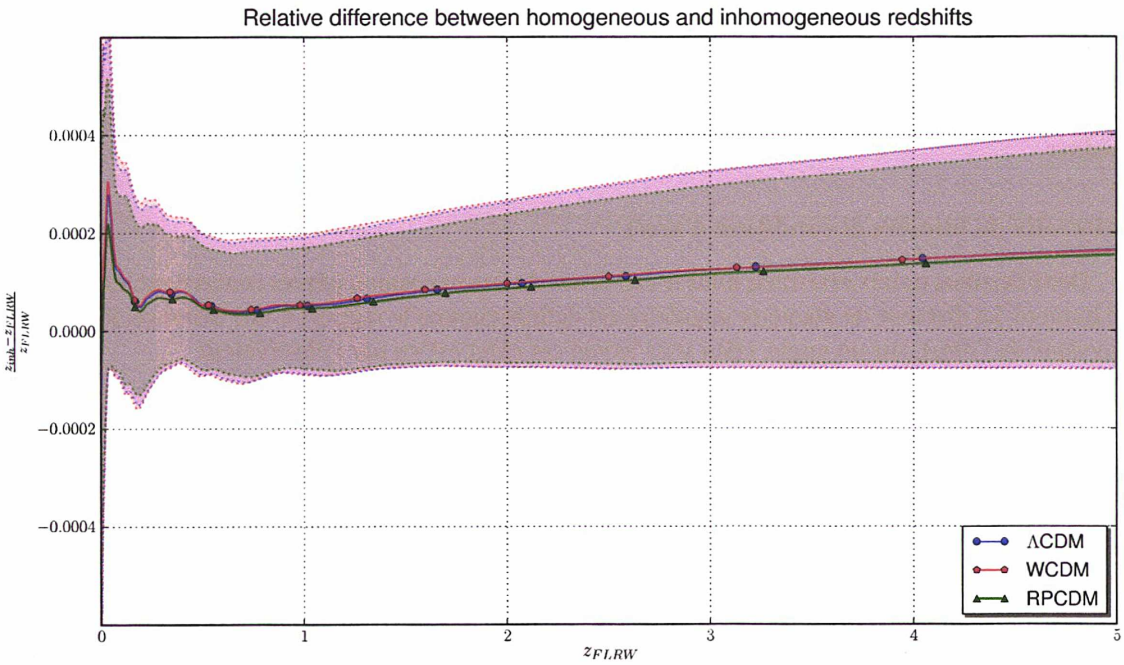


Figure 36 – Différence relative entre le redshift inhomogène et le redshift homogène. Pour chaque trajectoire, le redshift « inhomogène » calculé à l'aide de l'équation 4.11 est comparé au redshift à la même distance comobile pour le modèle homogène friedmannien. La moyenne est ensuite calculée pour un million de faisceaux sur tout le ciel, de même que l'écart-type. La première correspond aux traits pleins ci-dessous tandis que les domaines affichés par transparence correspondent aux seconds. Pour les trois cosmologies, la phase simulée est la même, et les rayons sont lancés dans les mêmes directions : la légère différence entre ΛCDM, WCDM et RPCDM est donc réellement de nature cosmologique. À bas redshift, l'écart entre le reshift homogène et le redshift inhomogène est de l'ordre de 2×10^{-4}, puis il se stabilise à environ 5×10^{-5} à $z = 1$ avant de réaugmenter progressivement vers 2×10^{-4} avec un écart-type du même ordre de grandeur.

acoustiques des baryons ou pour le fond diffus cosmologique. En supposant une cosmologie homogène toutes les définitions se rejoignent, mais dans le cas perturbé, chaque type d'observation a en réalité sa propre définition de la mesure des distances de diamètre angulaire. En explorant plusieurs définitions, les simulations numériques seraient à même de quantifier les effets des inhomogénéités pour chacune des définitions. Toutefois, il s'agit d'une question ouverte qui devra faire l'objet d'analyses plus détaillées.

4.2 Résultats

4.2.1 Effets sur le redshift

Une fois l'intégrateur créé et les tests de propagation décrits dans la partie numérique de ce manuscrit validés, le premier effet cosmologique mesurable est celui intervenant sur le redshift. Dans toute la suite, on se place dans le cadre du lancement d'un million de faisceaux, c'est-à-dire de huit millions de photons lancés dans des directions aléatoires sur toute la sphère céleste. Pour s'assurer de la robustesse des résultats, les mêmes opérations ont été effectués pour dix millions de faisceaux dans le cadre de ΛCDM : les valeurs des quantités statistiques calculées sont sensiblement les mêmes. Pour chaque faisceau, il est possible de comparer le redshift homogène résultant d'un calcul friedmannien au redshift inhomogène calculé à l'aide de la formule 4.11 et dépendant du rapport des potentiels locaux. Le résultat sur la moyenne et l'écart-type pour un observateur virtuel du Full Universe Run est illustré en figure 36. De brusques variations de redshift sont observées dans l'Univers proche avant d'atteindre des différences relatives entre le cas inhomogène et le cas homogène de l'ordre de 5×10^{-5} à $z = 1$. Des

faibles différences entre cosmologies apparaissent également, notamment entre ΛCDM et RPCDM. Le fait que la différence relative se stabilise sur des valeurs positives signifie que le redshift inhomogène d'une source S située à une distance comobile χ_S de l'observateur est légèrement supérieur au redshift homogène. Ou, vu dans l'autre sens, cela signifie que la distance comobile associée à un redshift interprété dans une cosmologie purement homogène est légèrement sous-estimée. Il y a donc un biais sur la mesure de $\chi(z)$ qui peut naturellement se traduire par une erreur d'appréciation des paramètres cosmologiques. Pour aller plus loin, il est maintenant nécessaire de se questionner sur l'impact couplé de ce biais sur le redshift et de la déformation des faisceaux lumineux sur la mesure des distances de diamètre angulaire.

4.2.2 Effets sur les distances cosmologiques

Pour mesurer la distance de diamètre angulaire déduite de l'intégration des géodésiques dans des univers structurés, on utilise la variation de rayon des faisceaux lumineux. En effet, du fait des inhomogénéités, le cercle de photons lancé initialement va avoir tendance à s'élargir ou se contracter suivant que l'Univers est localement sous-dense ou surdense. Pour s'assurer de la validité des conclusion on effectue les mesures pour deux observateurs en lançant à chaque fois un million de faisceaux dans toutes les directions. Dans le détail, il s'agit des observateurs 1 et 2 du Full Universe Run, le premier étant situé au centre de la boîte $(0.5, 0.5, 0.5)$, et le second aux coordonnées normalisées $(0.5, 0.1, 0.1)$. À chaque redshift inhomogène, c'est-à-dire calculé en tenant compte des potentiels locaux, on calcule :

$$d_A = \frac{r}{\theta} \tag{4.13}$$

où r est le rayon moyen du faisceau et où θ est l'angle de séparation des photons au niveau de l'observateur. Pour chaque modèle cosmologique, la moyenne et l'écart-type des distances de diamètre angulaire sont comparées au cas friedmannien. Les résultats obtenus correspondent aux courbes 37 et 38. Si les différences relatives sur le redshift étaient de l'ordre de 10^{-5} ou 10^{-4}, les différences relatives sur les moyennes des distances de diamètre angulaire sont beaucoup plus élevées. En effet, l'écart induit sur la moyenne par les inhomogénéités est de l'ordre de quelques pourcents. Au-delà de $z \approx 5$, celui-ci se stabilise du fait de la relative homogénéité de l'Univers. Bien que légères, des différences, liées en partie aux surdensités et aux sous-densités locales, existent tout de même pour les deux observateurs. Les écarts-types relatifs sont du même ordre de grandeur, sauf que, contrairement aux moyennes, ils sont mesurables observationnellement. Que ce soit pour la moyenne ou l'écart-type, les différences entre ΛCDM, WCDM et RPCDM sont bien visibles sur les figures : comme l'on pouvait s'y attendre, les effets sont d'autant plus grands que les modèles sont structurés.

Les figures 39 et 40 permettent de comparer les situations des modèles plus en détail. Pour les bas redshifts, en dessous de $z = 1$, les différences entre les cas homogènes et inhomogènes sont peu visibles du fait de la rapide augmentation de la distance de diamètre angulaire. Toutefois, près du pic à $z \approx 1.6$, la situation est bien différente. Il apparait que la structuration réhausse systématiquement les distances mesurées. Cela signifie qu'un modèle moins structuré comme RPCDM pourrait être interprété faussement comme un modèle de concordance rigoureusement homogène. Et le fait que la courbe en pointillés bleus de ΛCDM homogène rentre dans la zone à 1σ de la moyenne des distances mesurées dans RPCDM ne fait que confirmer cette possibilité. De même, une cosmologie de concordance pourrait être interprétée comme un modèle phantom. Finalement, en supposant la relation d'Etherington valide, il est possible d'obtenir les diagrammes de Hubble de la figure 41, c'est-à-dire la distance de luminosité en fonction du redshift. Les conclusions des figures précédentes demeurent. De futurs grands relevés observationnels de supernovæ pourraient venir contraindre la cosmologie via de tels diagrammes en rendant possible le calcul du spectre de puissance des fluctuations des distances de luminosité.

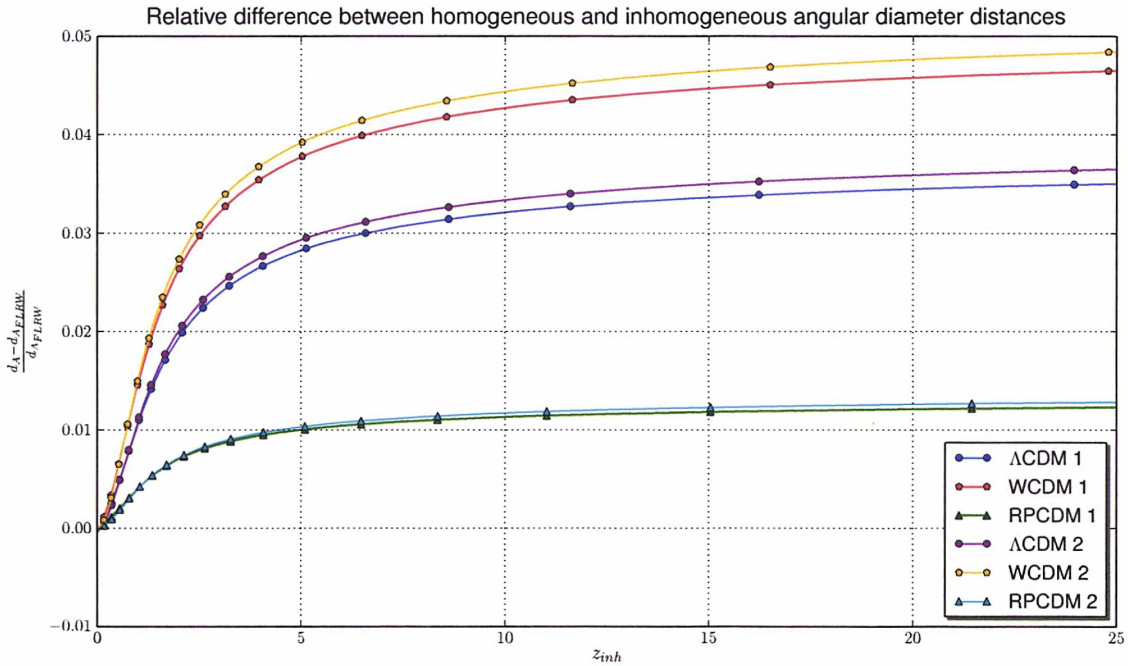


FIGURE 37 – Différence relative entre la distance de diamètre angulaire homogène et la distance de diamètre angulaire inhomogène moyenne jusqu'à $z = 25$. À chaque redshift on compare la valeur de la distance de diamètre angulaire friedmannienne homogène et celle calculée à partir de la variation de taille des faisceaux de lumière intégrés dans les simulations du Full Universe Run. Chaque modèle cosmologique est traité de façon indépendante, c'est-à-dire que sa référence correspond au modèle friedmannien associé à ses paramètres cosmologiques spécifiques. Les numéros 1 et 2 correspondent à deux observateurs différents. Il apparait que le biais dépend de la cosmologie : à $z = 5$, il est de l'ordre de 4% pour WCDM, de l'ordre de 1% pour ΛCDM, et de l'ordre de 1% pour RPCDM.

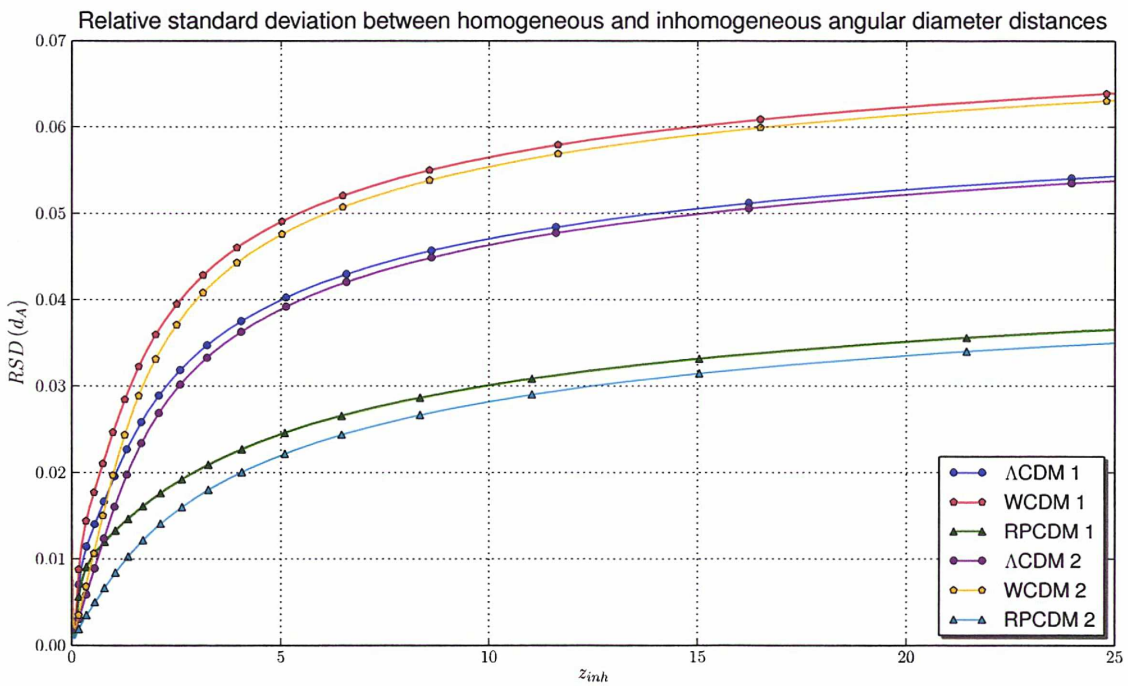


FIGURE 38 – Écart-type relatif (RSD) de la distance de diamètre angulaire inhomogène jusqu'à $z = 25$. Est illustré ci-dessus l'écart-type relatif $\frac{\sigma}{d_A}$ obtenu pour 1 million de faisceaux distribués sur tout le ciel pour deux observateurs dans les trois modèles cosmologiques. Contrairement à la moyenne qui correspond à des erreurs systématiques, l'écart-type est une grandeur observable. Pour ΛCDM il s'élève à environ 3% à $z = 2$ ce qui est loin d'être négligeable. Le fait qu'il soit dépendant de la cosmologie et qu'il soit mesurable en fait une sonde observationnelle.

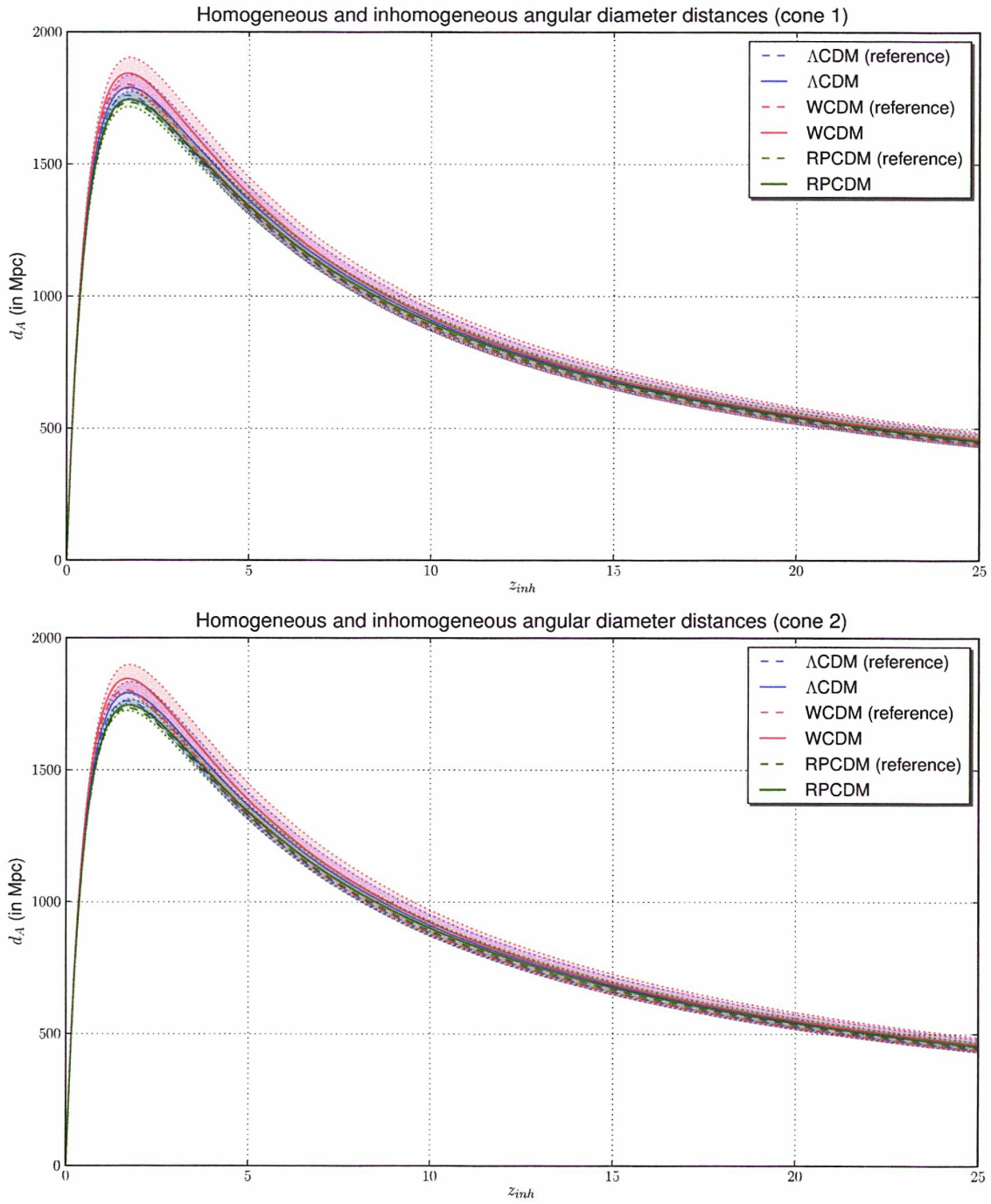


FIGURE 39 – Distances de diamètre angulaire homogènes et inhomogènes pour deux observateurs jusqu'à $z = 25$. En combinant les figures 37 et 38, on obtient les deux figures ci-dessus, chacune étant associée à un observateur. Les traits pointillés correspondent aux modèles homogènes, les traits pleins aux moyennes des distances de diamètre angulaire inhomogènes et les zones transparentes correspondent aux écart-types de part et d'autre des moyennes. Les différences les plus visibles apparaissent près du maximum de la distance de diamètre angulaire, c'est-à-dire à un redshift d'environ 1.6. À haut redshift, du fait de l'homogénéité du champ de matière, l'écart relatif des distances homogènes et inhomogènes se stabilise.

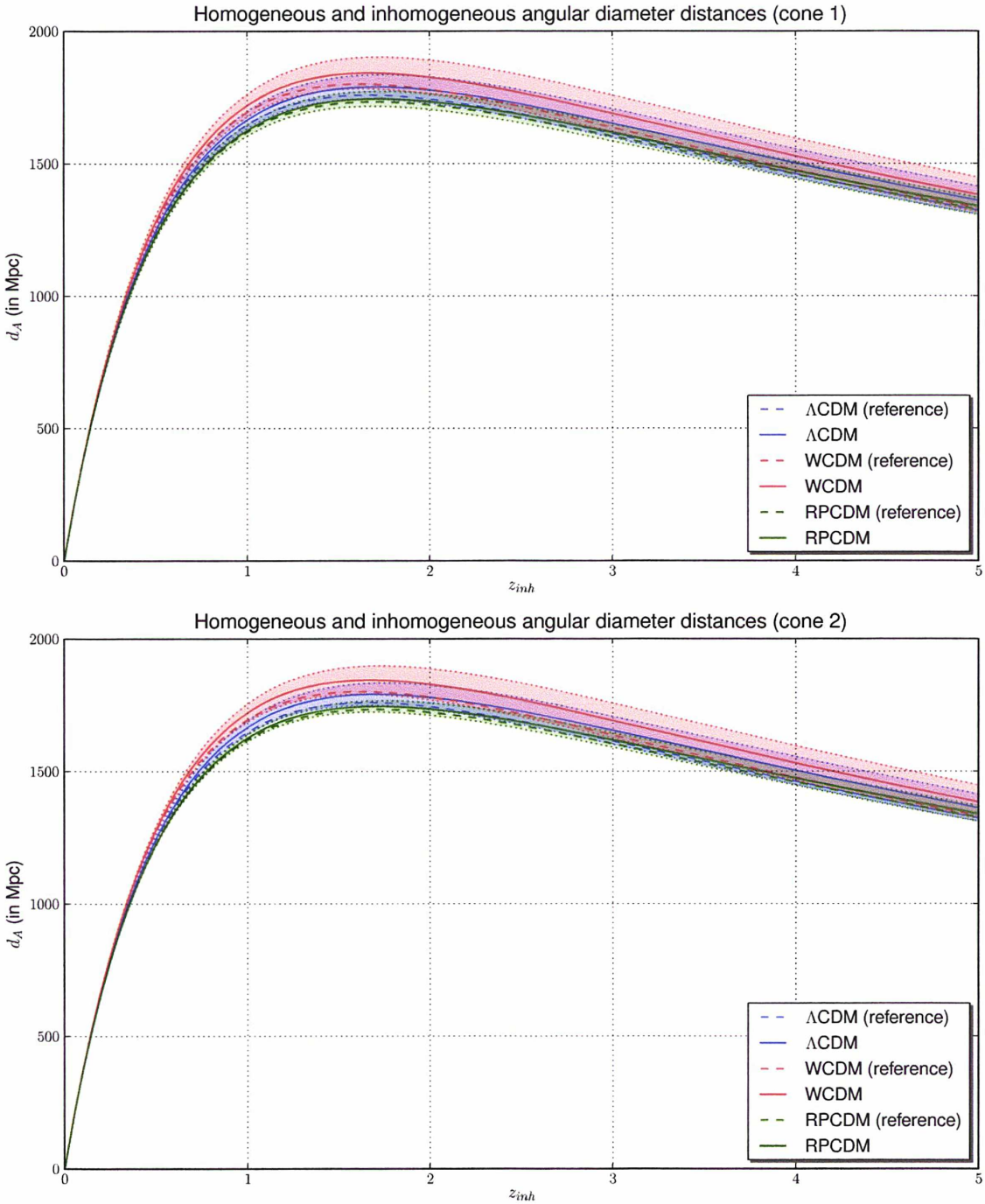


FIGURE 40 – Distances de diamètre angulaire homogènes et inhomogènes pour deux observateurs jusqu'à $z = 5$. Pour mieux mettre en évidence l'effet des inhomogénéités pour les redshifts observables, on agrandit la figure 39 dans le domaine $z \in [0, 5]$, c'est-à-dire dans la zone proche du maximum de distances de diamètre angulaire. Le fait qu'une distance homogène pénètre dans la zone à 1σ de la moyenne des distances inhomogènes d'un modèle différent signifie que le second modèle cosmologique pourrait être faussement interprété comme le premier. En l'occurence RPCDM pourrait être identifié comme ΛCDM si l'effet des inhomogénéités n'est pas pris en compte dans les analyses. La quantification de tout cela demanderait toutefois des analyses plus poussées pour mettre en évidence l'erreur induite par la distance de diamètre angulaire sur les paramètres cosmologiques.

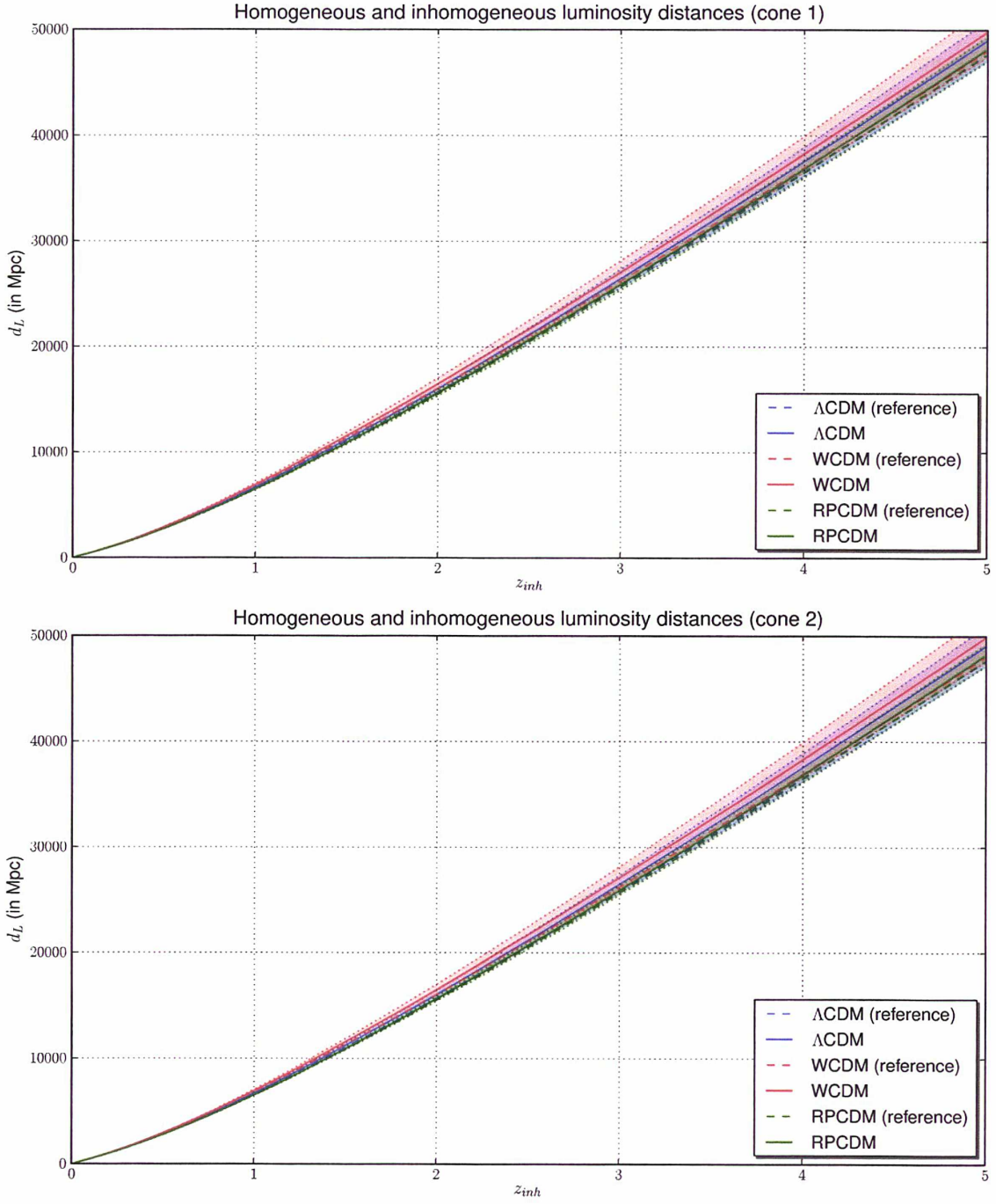


FIGURE 41 – Distances de luminosité homogènes et inhomogènes pour deux observateurs. En supposant valide la relation de réciprocité d'Etherington $d_L = (1+z)^2\, d_A$, il est possible d'obtenir directement le diagramme de Hubble, utilisé classiquement dans le cadre des relevés de supernovæ, à partir de la figure 40. La courbe de la distance de luminosité de ΛCDM homogène, marquée en pointillés bleus, est située entre celle de WCDM et de RPCDM. Les deux observateurs obtiennent des résultats de mesure comparables. L'effet des inhomogénéités a tendance a réhausser les valeurs de la distance de luminosité par rapport aux modèles friedmanniens, si bien qu'un modèle moins structuré comme RPCDM peut être interprété comme un modèle de concordance rigoureusement homogène.

4.3 Implications cosmologiques et perspectives

4.3.1 Bilan

Comme nous venons de le voir, et comme nous pouvions le pressentir du fait des différents niveaux de structuration des modèles de concordance, de Ratra-Peebles et d'énergie noire phantom, la cosmologie laisse son empreinte dans les distances de diamètre angulaire et les distances de luminosité, qui constituent toutes deux des points nodaux de nombre d'analyses standard. Deux effets peuvent être distingués : d'une part des effets sur les moyennes, d'autre part des effets sur l'écart-type. Les premiers ne sont pas directement observables et viennent perturber l'ensemble des observations quand il s'agit de déduire les paramètres cosmologiques de celles-ci. L'amplitude de ces perturbations est dépendant de l'histoire cosmique de formation des structures, donc de la cosmologie, et peut atteindre de l'ordre de 2% à $z = 2$ pour le modèle de concordance comme présenté en figure 37. Aussi, la réinterprétation dans un cadre purement homogène des distances de diamètre angulaire mesurées peut entrainer, de fait, des erreurs d'évaluation des paramètres cosmologiques. Les seconds, c'est-à-dire les effets sur l'écart-type sont eux, en revanche, observables. Là encore, ils dépendent de la cosmologie, comme illustré en figure 38 : environ 1.5% à $z = 2$ pour RPCDM, contre 2.5% pour ΛCDM et 3.5% pour WCDM. En conséquence, le spectre de puissance des fluctuations des distances de diamètre angulaire est une sonde pour la cosmologie dans l'espace des redshifts. Ou dit autrement, la lumière ne transporte pas seulement de l'information sur les objets observés : sa propagation elle-même garde l'empreinte de l'accélération de l'expansion et en ce sens est capable de nous apporter de nouveaux éléments sur cette dernière.

4.3.2 Perspectives

Les travaux présentés dans ce chapitre ouvrent de très nombreuses perspectives. Concernant la mesure des distances cosmologiques bien entendu, mais également et surtout concernant toutes les analyses permises par un traitement unifié de la propagation de la lumière. En effet, les résultats discutés ici peuvent être vus comme une application particulière d'une approche générique aux conséquences des inhomogénéités sur la distance de diamètre angulaire. En restant au plus près de la relativité générale et en évitant des simplifications qui seraient liées à un formalisme particulier, il est possible de traiter simultanément le lentillage faible, l'effet Sachs-Wolfe intégré ou la mesure des distances cosmologiques dans un univers structuré sur les architectures actuelles des supercalculateurs. De plus, le fait que ces machines soient maintenant capables de traiter la dynamique gravitationnelle à l'échelle de l'Univers Observable permet de se soustraire à bon nombre d'approximations usuelles quand il s'agit d'analyser des simulations dans l'espace des redshifts.

Tout d'abord la suite immédiate à donner à ce travail d'un point de vue physique est double. D'une part, il s'agit de quantifier l'incidence de la propagation de la lumière sur l'évaluation des paramètres cosmologiques Ω_M, Ω_Λ, w_{DE}..., par exemple au moyen d'une analyse Monte-Carlo, et d'étudier plus en détail le spectre de puissance des fluctuations des distances de diamètre angulaire qui constitue une véritable observable cosmologique. D'autre part, il s'agit d'explorer deux des approximations présentées en début de chapitre : la construction des cônes au moyen de coquilles sphériques parfaites et le critère d'arrêt des faisceaux. Pour le premier, disposer de l'évolution de la structuration à tous les pas de temps dans un sous-volume de la simulation donne une opportunité unique d'effectuer du raytracing en 4 dimensions, pour lequel le temps est traité de la même façon que l'espace lors de l'intégration des géodésiques. La différence entre cette méthode et la méthode plus classique de construction des cônes devrait être très faible, mais la quantifier de façon rigoureuse permettrait de s'en assurer définitivement. La question du critère d'arrêt, elle, nécessite d'être éclaircie. Cela passe par une étude comparative des différents critères possibles et leurs conséquences sur la mesure des distances. Dans ce cadre, il semblerait particulièrement à propos de ne pas supposer *a priori* la validité de la relation d'Etherington, mais de la vérifier directement par des mesures au sein des simulations, confirmant du

même coup la cohérence du choix du critère d'arrêt pour le calcul des distances de diamètre angulaire et pour le calcul des distances de luminosité. Une fois ces deux problématiques traitées, des extensions de cette étude sont envisageables. D'une part, les champs de vitesse locaux pourraient être pris en compte dans le calcul du redshift alors qu'ils sont considérés pour le moment comme des effets indépendants. D'autre part, il serait possible d'étendre le formalisme à l'ordre supérieur en vue d'intégrer des effets plus fins dans les équations géodésiques. Toutefois, dans ce cas, se pose la question de la nature des données d'entrée de l'algorithme d'intégration des géodésiques, car la compatibilité avec les simulations cosmologiques standard ne serait plus assurée. Enfin, d'un point de vue numérique, et étant donné que la méthode mise en œuvre permet de plonger récursivement dans un maillage raffiné, il semble important de quantifier les effets de résolution dans le cadre de cônes de lumière plus petits, extraits de boîtes de simulation de taille plus faible mais plus résolues. De telles simulations, en plus de mettre à disposition un potentiel doté d'une meilleure définition dans l'entourage des halos fourniraient la possibilité d'explorer des angles d'ouverture moindres pour les faisceaux. Dans ce cadre, il serait également pertinent de tester des schémas d'interpolation d'ordre plus élevé, du genre « Triangular Shaped Cloud », à même de garantir la continuité de la dérivée seconde du potentiel.

Au niveau d'autres sujets d'étude, on peut naturellement penser à celui du lentillage, notamment dans le cadre de la préparation de missions comme Euclid. En effet, l'approche « ray bundle » mise en place pour la propagation des faisceaux est tout à fait compatible avec le calcul de la convergence, du cisaillement, de la magnification et des autres quantités du formalisme du lentillage. Dans le cadre de problématiques cosmologiques l'accès à de très grands espaces de redshift au moyen des simulations numériques est essentiel, mais des analyses dans des espaces beaucoup plus restreints et beaucoup plus résolus de même que des analyses à partir de distributions de masse reconstruites sont envisageables en utilisant le même traitement et les mêmes codes que ceux utilisés pour obtenir les résultats de cette partie. Car la seule chose dont a besoin l'intégrateur pour résoudre les équations géodésiques, c'est d'une distribution tridimensionnelle de potentiel gravitationnel. Il serait également possible d'aller plus loin en généralisant encore un peu plus le traitement proposé ici. En effet, nous nous sommes placés dans le cadre du formalisme perturbatif en champ faible ce qui fait que la métrique était totalement spécifiée par la donnée de Ψ et de Φ. Or, l'on pourrait tout à fait imaginer adopter la même approche non pour une distribution arbitraire de potentiel, mais pour une distribution arbitraire de métrique. Autrement dit, au lieu de stocker le potentiel dans chaque cellule d'une simulation, le traitement des équations géodésiques s'appuierait sur le stockage de quantités tensorielles dans chacune des cellules. Numériquement, cela demanderait un espace de stockage bien plus grand, mais cela permettrait d'atteindre une généricité maximale. Dans ce contexte, la métrique en champ faible ne serait plus qu'un cas d'étude particulier et l'analyse de la propagation de la lumière pourrait être menée pour des métriques génériques non nécessairement diagonales.

4.3.3 Conclusion

Cette partie physique nous a mené de l'Univers homogène à l'Univers observé reconstruit en traitant la question de la propagation de la lumière au travers des grandes structures. Nous avons montré que différents modèles cosmologiques réalistes au sens de ce qu'ils sont compatibles avec les observations conduisaient à différents niveaux de structuration. Or cette structuration modifie la métrique locale ce qui affecte directement les géodésiques nulles, c'est-à-dire la propagation de la lumière. Dès lors, la question de savoir ce que nous disent vraiment les observations se pose.

Pour aborder cette question, nous avons considéré trois modèles cosmologiques : le modèle de concordance ΛCDM, un modèle de quintessence de Ratra-Peebles RPCDM et un modèle phantom WCDM. Pour ces trois modèles, des simulations de la taille de l'Univers Observable ont été réalisées dans le cadre du Full Universe Run. Cela a permis de sonder le spectre de puissance aux très grandes échelles et la fonction de masse pour des évènements extrêmes de plus de $10^{16}\,\mathrm{M}_\odot/\mathrm{h}$. Le fait que RPCDM aboutisse à un niveau de structuration moindre que le modèle de concordance tandis que

WCDM conduit à la formation de davantage de structures allait permettre d'explorer l'influence de la cosmologie sur la propagation de la lumière de part et d'autre de ΛCDM.

En vue d'être en mesure de traiter un maximum d'effets, une approche générique et unifiée du problème a été choisie. Pour cela les équations géodésiques dans le cadre d'une métrique faiblement perturbée ont été gardées sous leur forme la plus générale. D'autre part des cônes de lumière ont été construits pour chaque simulation, et pour plusieurs observateurs virtuels par simulation en évitant un certain nombre d'approximations usuelles. Dès lors, disposant de la distribution de potentiel gravitationnel dans tout l'espace pour plusieurs cosmologies, il devenait possible de propager des photons uniques et des faisceaux de lumière. Cela a permis de mettre en évidence l'effet de la cosmologie sur la mesure des distances de diamètre angulaire. Aussi, interpréter une mesure de distance comme si la cosmologie était rigoureusement homogène peut conduire à une mauvaise estimation des paramètres cosmologiques. Cet effet a été quantifié pour les différents modèles et il est loin d'être négligeable. Il est par ailleurs d'autant plus grand que l'Univers est structuré. Ainsi, le spectre de puissance des fluctuations de distances de diamètre angulaire ou de distances de luminosité, est lui-même une sonde pour la cosmologie.

Le traitement de la propagation de la lumière proposé dans ces pages ouvre la porte à de multiples analyses cosmologiques. Les quelques résultats présentés dans ce dernier chapitre peuvent être vus comme une façon de valider la pertinence de l'approche. En restant au plus proche des principes fondamentaux, le numérique permet de traiter à l'aide d'un seul formalisme et d'un seul code numérique de très nombreuses problématiques. En l'occurence, l'intégration des géodésiques nulles dans une distribution arbitraire de potentiel peut permettre d'aborder des problématiques de mesure de distance, des problématiques de lentillage gravitationnel ou des problématiques ayant trait à l'effet Sachs-Wolfe intégré.

D'un point de vue physique, le fait que les inhomogénéités ont une influence sur la façon dont les observations doivent être traitées et analysées est de première importance à l'heure où les propriétés de l'accélération de l'expansion sont sondées d'une manière de plus en plus précise et par le biais de données indépendantes de plus en plus nombreuses. La cosmologie physique ne peut plus faire abstraction du fait que, localement, l'Univers n'est pas homogène. Nous avons montré que cela devait être pris en compte pour les géodésiques nulles. Dès lors, cela ne peut que renforcer la nécessité d'explorer la question ouverte du cas des géodésiques non-nulles, et du cas de la dynamique de la matière dans un cadre localement inhomogène.

Chapitre 5

Vers de nouvelles architectures logicielles pour la cosmologie numérique

***

5.1 Cosmologie numérique et simulations cosmologiques . . . 90
5.1.1 Contexte et caractéristiques des simulations . . . 90
5.1.2 Déroulement des simulations . . . 93
5.2 Le projet Dark Energy Universe Simulation . . . 94
5.2.1 Problématiques et objectifs . . . 94
5.2.2 Codes et procédures . . . 94
5.3 Le Full Universe Run : une expérience numérique . . . 95
5.3.1 Configuration . . . 95
5.3.2 Temps de calcul et communications . . . 98
5.3.3 Mémoire . . . 98
5.3.4 Entrées et sorties . . . 101
5.3.5 Pré et post-traitements . . . 103
5.4 Limitations de la cosmologie numérique . . . 103
5.4.1 Des succès aux limitations . . . 103
5.4.2 Leçons du Full Universe Run . . . 104
5.4.3 À la source de tous les maux : l'intrication . . . 107
5.4.4 Problématique . . . 108
5.5 Programmation générique, programmation générative et métaprogrammation . . . 110
5.5.1 Choix techniques . . . 110
5.5.2 Programmation générique et factorisation de code . . . 111
5.5.3 Métaprogrammation . . . 114
5.6 Application à la cosmologie . . . 121
5.6.1 Vers des bibliothèques actives pour la cosmologie : l'exemple de Magrathea . . . 121
5.6.2 Proposition d'architecture logicielle unifiée pour la cosmologie numérique . . . 127

Là où se termine le domaine du linéaire et des approches perturbatives débute le domaine de la simulation numérique. En effet, comme cela a été déjà souligné dans l'avant-propos, la simulation numérique est à même de déconstruire les symétries imposées par les traitements analytiques. Depuis les

années 1970 et les premières simulations de quelques centaines de particules en interaction gravitationnelle, la simulation numérique est devenue un outil de premier plan pour traiter les phénomènes non-linéaires tels que la formation des structures cosmologiques. À mesure que la puissance de calcul progressait, il devenait possible d'explorer des échelles de plus en plus grandes ou d'explorer la dynamique gravitationnelle avec de plus en plus de résolution. Dans cette optique, les premiers codes se sont peu à peu étoffés, faisant appel à des méthodes numériques de plus en plus évoluées, à des algorithmes de parallélisation de plus en plus optimisés et à une physique de plus en plus riche. Ce faisant, des codes de plus en plus complexes ont vu le jour.

L'objectif de ce chapitre à dominante numérique est de proposer une nouvelle façon de concevoir les codes de simulation cosmologique pour que la physique puisse s'y exprimer le plus naturellement possible. Les trois premières sections nous conduiront au travers de la réalisation du Full Universe Run, premier ensemble de simulations de la structuration à l'échelle de l'Univers Observable. Les trois suivantes, plus techniques, car il s'agira d'explorer et de mettre en œuvre de nouveaux concepts, nous permettront de tirer toutes les leçons de cette expérience numérique en nous réinterrogeant sur les codes de simulations eux-mêmes. Nous proposerons ainsi de nouvelles approches basées sur des bibliothèques actives et des langages embarqués pour explorer une physique toujours plus pertinente sur des architectures toujours plus complexes.

La première section rappelle les principes de base de ces simulations ainsi que les méthodes numériques les plus utilisées dans le domaine. La seconde se penche sur le projet Dark Energy Universe Simulation au travers de ses trois composantes : les Dark Energy Universe Simulation Series (DEUSS), destinées à explorer l'espace des paramètres physiques et numériques, le Full Universe Run (DEUS-FUR), destiné à sonder la structuration à l'échelle de l'Univers Observable, et le Parallel Universe Run (DEUS-PUR), destiné à tester l'hypothèse ergodique au travers de la réalisation d'un très grand nombre de simulations. La troisième section détaille la réalisation du Full Universe Run sur la machine Curie en 2012, qui a permis de mettre en exergue les succès actuels de la cosmologie numérique mais aussi un certain nombre de problèmes structurels. Ces problèmes et les limitations qu'ils entrainent sont abordés dans la quatrième section qui s'attache à poser les bonnes questions concernant la conception logicielle. La cinquième partie présente un ensemble de solutions à ces problèmes en se basant sur de la métaprogrammation par template. Enfin, la dernière partie décrit l'application de tous ces concepts à l'implémentation d'une nouvelle bibliothèque pour la cosmologie numérique mise en place dans le cadre de cette thèse, Magrathea et propose une architecture logicielle unifiée pour la cosmologie numérique. Les deux chapitres suivants concerneront la mise en application de tous ces principes dans le cadre du suivi de la propagation de la lumière dans un univers perturbé. On rappelle que dans ce chapitre, on se concentre sur les aspects numériques des simulations, les aspects physiques en lien avec cette thèse ayant été décrits dans les chapitres de la partie précédente.

5.1 Cosmologie numérique et simulations cosmologiques

5.1.1 Contexte et caractéristiques des simulations

Hypothèses

Pour étudier les processus non-linéaires de formation des structures, les simulations cosmologiques reposent sur plusieurs hypothèses. Tout d'abord, on se place dans le cadre du principe cosmologique à grande échelle pour la métrique. Ce faisant, on néglige les rétroactions de la structuration sur l'expansion, ce qui conduit à un traitement newtonien de l'interaction gravitationnelle en coordonnées comobiles. Ainsi, l'évolution du facteur d'échelle en fonction du temps cosmique peut être considérée comme une entrée des codes dynamiques chargés de traiter l'effondrement des structures. Ensuite, l'espace des phases associé au champ de matière doit être discrétisé. Les codes dynamiques peuvent s'appuyer sur deux approches : l'une lagrangienne à base de particules, l'autre eulérienne à base d'un découpage de

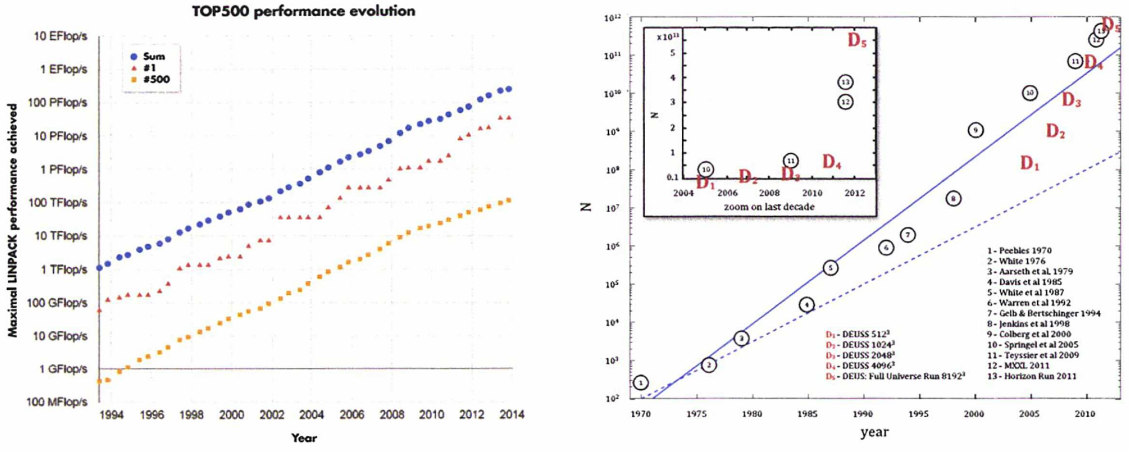
l'espace en cellules et d'une évaluation des flux aux interfaces de ces dernières. Aux larges échelles, le problème peut être traité sous l'angle purement gravitationnel, en négligeant les problématiques hydrodynamiques : l'approche lagrangienne des codes N-corps est alors tout à fait adaptée. Plutôt que de résoudre le problème N-corps dans l'espace physique, notons qu'il est également possible d'intégrer directement le système d'équations de Vlasov-Poisson dans l'espace des phases 6-dimensionnel[111]. Pour tester la pertinence des modèles cosmologiques, les conditions initiales sont basées sur des réalisations particulières, c'est-à-dire des phases, des spectres de puissance initiaux à z_{CMB} évolués via l'approximation de Zel'dovich ou via la théorie perturbative lagrangienne à des ordres supérieurs jusqu'au redshift de départ z_{ini}. Ce dernier doit être choisi ni trop élevé, auquel cas des erreurs numériques s'accumulent dans l'histoire primitive de l'Univers, ni trop faible, auquel cas l'approximation de Zel'dovich n'est plus valide. Lorsque l'on cherche à reproduire un environnement particulier il est possible, soit de travailler avec des phases contraintes, soit d'effectuer des simulations « zoom » en resimulant certaines zones d'une simulation grossière avec davantage de résolution. Ces techniques, mentionnées à titre indicatif, ne seront toutefois pas employées dans le cadre du projet Dark Energy Universe Simulation décrit dans ce chapitre. Enfin, au niveau des conditions au bord et de la topologie, on considère un tore tridimensionnel en simulant un cube avec des conditions aux limites périodiques. Cela est motivé par le fait que l'Univers tend vers l'homogénéité aux grandes échelles.

Propriétés

Comparativement à d'autres domaines très demandeurs de simulation numérique, les simulations cosmologiques comportent un certain nombre de caractéristiques propres. Tout d'abord, et c'est peut-être le point le plus important, la complexité des simulations ne réside pas dans la géométrie des maillages, comme cela est usuellement le cas dans les simulations de mécanique des solides par exemple. La géométrie est très simple puisqu'il s'agit d'une boîte de simulation traitée en coordonnées cartésiennes. De plus, à part dans le cas de simulations « zoom » ou de simulations galactiques à une échelle plus réduite, c'est toute la boîte qui a un intérêt. Cela diffère par exemple des simulations CFD (Computational Fluid Dynamics) au sein de systèmes industriels, ou plus proche de l'astrophysique, des simulations relativistes d'objets compacts. Mais s'intéresser à toute la boîte ne signifie pas pour autant que la résolution doit être uniforme. En effet, à mesure que l'Univers s'expand, les structures se forment et il est de première importance d'être à même de résoudre les régions surdenses. C'est pour cette raison que les maillages adaptatifs basés sur des raffinements ou sur de la tessellation sont couramment utilisés dans ce domaine d'application : ces derniers permettent en effet d'atteindre de meilleures résolutions spatiales que dans le cadre de grilles fixes. La résolution en masse, elle, dans le cas d'un solveur N-corps, dépend du nombre de particules. Du fait de ces deux résolutions, spatiale et massique, les simulations cosmologiques peuvent être caractérisées par deux échelles : la première est la taille de l'Univers Observable, d'un diamètre de l'ordre de 21 000 $\mathrm{Mpc/h}$, tandis que la seconde est la masse minimum des halos à formation d'étoiles, de l'ordre de $10^7\ \mathrm{M_\odot/h}$[112], sachant qu'un halo peut être considéré comme résolu lorsqu'il est constitué d'une centaine de particules. En termes purement numériques, une simulation cosmologique mêlant ces deux échelles correspondrait à une simulation hydrodynamique de la structuration dans un volume de 21 $\mathrm{Gpc/h}$ avec des particules de $10^5\ \mathrm{M_\odot/h}$. Cela reste encore complètement hors de portée des machines actuelles.

Évolution

L'évolution de la cosmologie numérique est liée d'une part à l'augmentation des performances des supercalculateurs et d'autre part à l'amélioration des algorithmes et à l'optimisation des codes, comme illustré en figure 42. En 20 ans, la machine classée première au TOP500 est passée d'environ 140 $\mathrm{GFlop/s}$ à près de 34 $\mathrm{PFlop/s}$ soit une augmentation d'un facteur de l'ordre de 250 000 . L'évaluation des performances d'une machine à l'aune d'un test d'algèbre linéaire peut conduire à une surestimation des capacités de calcul comparativement à celles obtenues dans le cadre d'une application réelle, beaucoup plus complexe, mais elle donne néanmoins un aperçu des performances brutes des



(a) Augmentation des performances des supercalculateurs du TOP500.

(b) Évolution du nombre de particules au sein des simulations cosmologiques

Figure 42 – Évolution conjointe de la puissance des supercalculateurs et du nombre de particules dans les simulations. À gauche, l'augmentation des performances des supercalculateurs du TOP500[113] depuis les années 1990. Ce classement répertorie les meilleurs performances obtenues avec le benchmark d'algèbre linéaire LINPACK[114] sur les supercalculateurs du monde entier. Deux fois par an, trois valeurs sont indiquées : la somme de la capacité de calcul de l'ensemble des 500 machines classées, en bleu, les performances de la première machine du classement, en rouge, et de la dernière, en jaune. Á ce rythme, une machine exaflopique devrait voir le jour en 2019. Toutefois, l'augmentation du coût des machines, de leur consommation électrique et les difficultés rencontrées par les fondeurs à diminuer la finesse de gravure des processeurs en dessous de 10 nm pour des raisons physiques pourraient ralentir la progression observée depuis 20 ans. Cette augmentation de la capacité de calcul s'est accompagnée d'une augmentation du nombre de particules utilisées pour les simulations cosmologiques, comme présenté à droite. Ce graphique, non exhaustif, répertorie quelques simulations marquantes depuis les années 1970. La courbe en pointillés bleus est calquée sur la loi de Moore et correspond à un doublement du nombre de particules tous les 18 mois. Le trait plein représente l'évolution moyenne du nombre de particules au sein des simulations[115]. Les « D » rouges repèrent les simulations du projet Dark Energy Universe Simulation, avec le Full Universe Run, en 2012. En 40 ans, les simulations sont passées de quelques centaines de particules, à plusieurs centaines de milliards.

machines. Sur la même durée, le nombre de particules au sein des simulations cosmologiques est passé typiquement de 128^3 à 8192^3 soit une augmentation d'un facteur 250 000. Mais l'augmentation de la puissance de calcul brute ne suffit pas à caractériser l'ensemble des ressources nécessaires pour les simulations : c'est d'ailleurs pour cela qu'il faut distinguer l'exaflop, qui ne concerne que le nombre d'opérations par seconde, de l'exascale qui concerne également les communications, la mémoire et la capacité de stockage. Si l'on se concentre sur les dix dernières années et que l'on compare le plus puissant supercalculateur actuel, Tianhe-2, installé en Chine, au plus puissant supercalculateur des années 2001 à 2004, le Earth Simulator installé au Japon, la puissance de calcul a été multipliée par 10^3, la quantité de mémoire par 10^2 et le stockage par 10 : toutes les composantes ne progressent donc pas à la même vitesse, ce qui sera un point capital pour la suite. À moins d'une rupture technologique majeure dans les années à venir, la puissance de calcul risque de continuer à augmenter plus rapidement que la mémoire disponible : or c'est cette dernière qui contraint en grande partie la résolution. Le développement de codes et d'algorithmes pour les machines exascale doit prendre en compte ces considérations pour tirer le meilleur parti des futurs supercalculateurs.

Ces progrès de la cosmologie numérique portés par les progrès matériels se sont accompagnés de progrès plus fondamentaux sur les méthodes numériques et la physique étudiée. Au niveau des premières, on pourrait citer : le calcul de toutes les interactions particules-particules (PP), les codes Particle-Mesh (PM), les codes Particle-Particle Particle-Mesh (P3M), les Tree codes et les Tree-PM. À cela on peut ajouter les améliorations fournies par les maillages adaptatifs[116] (PM-AMR) et par les algorithmes de tesselation[117, 118]. Du côté de la physique, on peut mentionner l'ajout de la physique baryonique, de la magnétohydrodynamique, la prise en compte du rayonnement, la possibilité de traiter

différents modèles d'énergie noire et différents modèles de gravité modifiée. La physique pouvant être prise en compte dans les simulations s'est donc considérablement enrichie au fil des ans, au moyen de codes toujours plus nombreux et toujours plus complexes. Les problèmes liés à cette augmentation de la complexité seront traités dans le prochain chapitre.

5.1.2 Déroulement des simulations

Pré-traitement

Les grandes lignes de la réalisation d'une simulation cosmologique ont déjà été décrites dans la partie physique. D'un point de vue numérique, une simulation peut être découpée en trois grandes phases : un pré-traitement, une résolution de la dynamique et un post-traitement. Le pré-traitement concerne la génération des conditions initiales et le calcul de la dynamique homogène si nécessaire. Par exemple, dans le cas d'un problème N-corps, il va s'agir de distribuer N points dans l'espace pour amorcer le calcul dynamique. Mais il ne s'agit pas de le faire n'importe comment : si l'on souhaite obtenir une structuration réaliste, il convient de s'assurer que le spectre initial de la distribution de matière est en accord avec le spectre du fond diffus cosmologique évolué linéairement jusqu'au redshift de départ. Un code s'occupe donc généralement de construire une réalisation particulière de la distribution de matière à partir du spectre correspondant au modèle cosmologique étudié. En fait, dans la pratique, cela revient à déplacer toutes les particules distribuées de façon homogène d'une quantité $\vec{\epsilon_i}$ et de leur affecter une vitesse $\vec{v_i}$, cette dernière étant souvent traitée comme une fonction de $\vec{\epsilon_i}$. Si besoin est, ces fichiers de conditions initiales sont ensuite redécoupés en vue d'optimiser la lecture parallèle par le code dynamique. En plus de cela, si le code dynamique ne prend pas en charge le calcul de l'évolution homogène, celle-ci doit être précalculée de façon séparée. Enfin, la physique est souvent spécifiée par un certain nombre de paramètres dont la liste peut s'accroître significativement lors de l'utilisation de modèles sous-grilles. Et plus la liste est longue, plus il est difficile de démêler l'influence des différents paramètres.

Dynamique

À partir des positions des particules, des tables d'évolution cosmologique et des listes de paramètres physiques, le code dynamique peut travailler. Numériquement, il s'agit souvent de la partie la plus lourde. Pas de temps, par pas de temps, la dynamique de chaque particule est intégrée en fonction des forces qui s'exercent sur elle. De nombreuses méthodes numériques existent pour calculer ces forces et en déduire l'évolution de la position des particules. Dans certains codes, de la même manière que la résolution spatiale peut être améliorée au moyen de maillages adaptatifs, la résolution temporelle peut être améliorée pour les mêmes conditions : il y a un sous-cyclage des pas de temps. Dans le cas spatial, comme dans le cas temporel, le niveau le moins résolu est appelé niveau grossier ou en anglais *coarse*. À certains pas de temps définis à l'avance, la simulation est sauvegardée. Ces sauvegardes ont deux objectifs : d'une part permettre un redémarrage du code en cas de plantage du supercalculateur sur lequel est effectuée la simulation, d'autre part extraire toutes les variables physiques pertinentes en vue des analyses.

Post-traitement

Enfin, une fois la phase dynamique terminée vient la phase de post-traitement, c'est-à-dire la phase d'analyses. Dans certain cas, toute ou partie de ces analyses peuvent être effectuées par le code dynamique lui-même : certains codes incluent ainsi des algorithmes de détection des structures « à la volée ». Toutefois, si cela est appréciable, cela conduit également souvent à une augmentation de la mémoire nécessaire. Dans certains cas, cela peut donc impacter les performances du code dynamique, surtout pour des simulations qui ont été spécifiquement dimensionnées pour atteindre une résolution maximale sur une machine donnée, comme ce sera le cas dans la suite. L'approche inverse consiste à

réaliser un maximum d'analyses une fois la simulation terminée. Comparativement à l'approche précédente, celle-ci peut se traduire par la nécessité de disposer d'un grand nombre de sauvegardes, donc par une occupation de l'espace disque plus importante. Mais disposer d'enregistrements génériques permet également d'effectuer ultérieurement des analyses qui n'avaient pas été prévues au départ. Reste une dernière possibilité, à mi-chemin entre les deux : lorsqu'il est impossible d'utiliser de la mémoire supplémentaire pour réaliser les analyses à la volée et lorsque l'espace disque vient à manquer, une stratégie alternative consiste à arrêter régulièrement le calcul dynamique pour effectuer toutes les analyses nécessaires sur un certain nombre de pas de temps avant d'effacer tous les fichiers temporaires et de relancer le calcul dynamique. A l'échelle pétaflopique, l'analyse de plusieurs téraoctets de données ne consiste souvent pas en un simple dépouillement facilement implémentable avec un langage de script : beaucoup d'analyses sont lourdes, demandent des codes parallèles qui peuvent parfois être plus complexes que les codes de simulation eux-mêmes. Pour répondre à cette problématique, il est possible d'effectuer les analyses en deux temps à l'aide d'un premier code parallèle compilé et optimisé capable d'extraire de l'information physique pertinente à partir des données brutes et d'un second niveau, éventuellement dans un langage de script, pour visualiser, filtrer, ou calculer des quantités annexes à partir des résultats du premier niveau.

5.2 Le projet Dark Energy Universe Simulation

5.2.1 Problématiques et objectifs

Intéressons nous maintenant au projet dans le cadre duquel a été effectué ce travail de thèse : le projet Dark Energy Universe Simulation. Ce projet, à la fois théorique et numérique, vise à fournir, entre autres choses, des éléments de réponse aux questions suivantes : quelles empreintes l'énergie noire laisse-t-elle dans la formation des grandes structures ? Et inversement, comment contraindre la nature de l'énergie noire à partir de l'observation des structures cosmiques ?

La simulation numérique telle que décrite dans les paragraphes précédents est un outil parfaitement adapté et nécessaire pour répondre à ces questions car elle permet d'explorer les processus non-linéaires et leurs conséquences observables pour une vaste gamme de modèles cosmologiques. Aussi, ce projet met à profit la puissance des supercalculateurs nationaux pour explorer trois voies différentes :

- l'exploration de l'espace des paramètres numériques et physiques
- l'exploration de la structure à très grande échelle et de la formation des halos les plus massifs pour différentes cosmologies au moyen de simulations de la taille de l'Univers Observable
- l'exploration de l'hypothèse ergodique et des erreurs statistiques au moyen d'un très grand nombre de simulations de petite taille mais d'un volume total comparable aux simulations du point précédent

Il s'agit là de trois voies standard empruntées lorsqu'il s'agit d'utiliser des ressources de calcul toujours plus étendues : soit l'on explore un espace des paramètres, qu'ils soient physiques ou numériques, plus grand, soit l'on utilise ces ressources pour disposer de davantage de résolution ou de statistique, soit, enfin, on met à profit cette puissance pour effectuer un grand nombre de simulations de taille raisonnable en parallèle.

5.2.2 Codes et procédures

Les trois pistes citées précédemment ont été explorées via trois composantes du projet :

- Dark Energy Universe Simulation Series (DEUSS) pour l'exploration de l'espace des paramètres physiques et numériques

- Le Full Universe Run (DEUS-FUR) pour l'exploration de simulations de la taille de l'Univers Observable
- Le Parallel Universe Run (DEUS-PUR) pour l'exploration de l'hypothèse ergodique et des erreurs statistiques

Les propriétés physiques des modèles cosmologiques explorés ont été données au chapitre premier. Dans tous les cas, c'est le code Ramses qui a été utilisé comme base pour effectuer les simulations avec divers degrés de modification et d'optimisation. Le Full Universe Run[36] a consisté en la réalisation de trois simulations de 8192^3 particules et de 21 Gpc/h de côté pour trois cosmologies différentes, tandis que le Parallel Universe Run[119] a consisté en la réalisation de trois ensembles de 4096 simulations de 256^3 particules et de 656.25 Mpc/h de côté ainsi que de 512 simulations de 512^3 particules et de 1312.5 Mpc/h de côté. Un résumé des caractéristiques de chaque ensemble de simulation est fourni en figures 44 et 45.

La réalisation de telles simulations passe par la conception d'un « workflow » de codes qui ne peut se résumer à l'utilisation d'un seul code dynamique : la génération des conditions initiales de même que l'analyse des données et la gestion des fichiers constituent autant de défis numériques annexes. Le workflow Amadeus conçu dans le cadre du Full Universe Run est résumé graphiquement en figure 43. Ce dernier s'appuie notamment sur les codes : MPGRAFIC[120] pour la gestion des conditions initiales, Ramses[116] pour le calcul dynamique, Powergrid pour le calcul des spectres de puissance et PFOF[66] pour la détection de structures et la gestion des données. Pour le Parallel Universe Run et la réalisation de plus de 12 000 simulations, ce workflow a été automatisé à l'aide de scripts. En effet, la réalisation d'un tel nombre de simulations ne peut être géré « manuellement ». Dans ce cadre, deux niveaux de scripts ont été conçus : le premier dédié à la gestion des travaux, le second à la réalisation et la surveillance des simulations. Ainsi, le premier niveau s'assurait du remplissage permanent de la queue de soumission sur la machine, tandis que le second automatisait tout le processus de la génération des conditions initiales au post-traitement. Ce second niveau incluait une vérification des fichiers produits en terme de nombre et de volume en vue de faciliter la détection des erreurs.

La réalisation des « Grands Challenges » que sont le Full Universe Run et le Parallel Universe Run ont constitué un travail important de cette thèse incluant le développement d'outils spécifiques comme les routines de construction des cônes de lumière présentées au chapitre quatre, la conception de scripts d'automatisation pour gérer plus de 12 000 simulations, le portage des codes sur les supercalculateurs Curie et Ada, leur optimisation via la réalisation de benchmarks préalables, la mise en œuvre et l'exécution du workflow Amadeus et la gestion, l'archivage et l'analyse des données produites.

5.3 Le Full Universe Run : une expérience numérique

5.3.1 Configuration

Pour comprendre les chapitres suivants et les problèmes qu'ils abordent, il est maintenant important de décrire les différentes optimisations qui ont été mises en place pour ce projet. Car, de ces optimisations découlent un certain nombre de leçons quant au portage de codes cosmologiques à l'échelle du petascale. Si atteindre l'échelle de l'Univers Observable avec des simulations marque, dans un certain sens, une étape dans l'évolution de la cosmologie numérique, elle est aussi l'occasion de prendre du recul vis-à-vis des codes eux-mêmes. L'« expérience »numérique dont il est question dans le titre de cette section fait donc référence à l'expérience accumulée lors du Full Universe Run.

DEUS-FUR a été dimensionné pour résoudre à la fois la formation des structures et contenir la taille de l'Univers Observable. Cela a nécessité la totalité du supercalculateur Curie, que ce soit du point de vue du temps de calcul, des communications, de la mémoire ou des entrées-sorties. Tous les aspects numériques étaient donc susceptibles d'être utilisés au maximum de leur performance. La conséquence

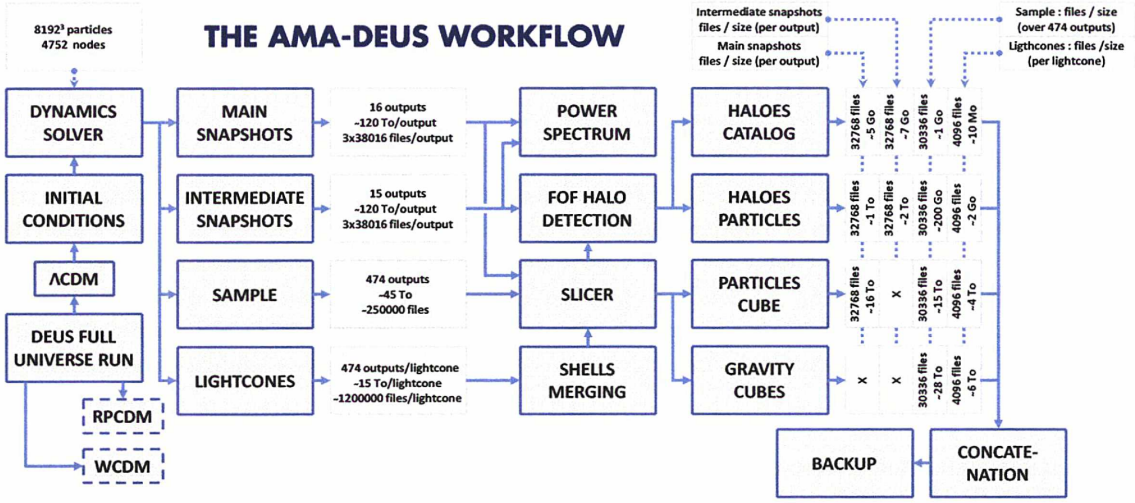


FIGURE 43 – Workflow utilisé pour la réalisation du Full Universe Run. Parce qu'il est impossible de tout sauvegarder pour des raisons d'espace disque et parce qu'il est impossible de tout analyser « à la volée » en local pour des raisons d'empreinte mémoire, la création d'un workflow de codes s'est avéré essentiel. Pour chaque simulation, tout commence par la génération des données initiales qui sont ensuite injectées dans le code dynamique. Celui-ci génère quatre types de données brutes : 16 enregistrements principaux, 15 enregistrements intermédiaires, un extrait de la boîte de simulation à chaque pas de temps ainsi que des coquilles de données nécessaires à la construction de cônes de lumière. Ces données brutes sont ensuite traitées par quatre programmes : l'un se charge de calculer les spectres de puissance, le second détecte les structures, le troisième redécoupe les données brutes en zones cubiques en vue de garantir une plus grande facilité d'analyse et le quatrième est chargé de fusionner les coquilles pour produire les cônes. Le volume des fichiers post-traités est indiqué dans la partie droite de la figure. Ces derniers sont finalement regroupés puis archivés sur les machines de calcul.

directe est que l'ensemble des codes d'AMADEUS a nécessité des adaptations et des optimisations spécifiques au projet. Dans ce contexte, avant de se pencher sur chacun des aspects numériques, il est important de décrire la configuration matérielle de la machine CURIE présentée en figure 47.

CURIE est le second système petascale du projet européen PRACE, géré par GENCI et installé au Très Grand Centre de Calcul du CEA. Les performances brutes de la machine au TOP500 sont de l'ordre de 2 PFlop/s [1]. Le système est hétérogène dans le sens où il comprend 3 partitions de calcul :

- 90 noeuds larges comprenant chacun 2 processeurs octocœurs INTEL NEHALEM-EX cadencés à 2.26 Ghz, 512 Gio de mémoire DDR3 à 4 Gio/coeur et 4 connexions INFINIBAND CONNECTX QDR
- 5040 nœuds fins BULLX B510 disposant chacun de 2 processeurs INTEL SANDYBRIDGE octocœurs cadencés à 2.7 Ghz, 64 Gio de mémoire DDR3 à 4 Gio/coeur, un disque local SSD et une connexion INFINIBAND CONNECTX QDR
- 144 nœuds hybrides BULLX B505 chacun doté de 2 processeurs quadricœurs INTEL WESTMERE cadencés à 2.67 Ghz et 2 accélérateurs graphiques NVIDIA MP2090

L'un des avantages de la machine du point de vue des simulations cosmologiques est de disposer d'une quantité élevée de mémoire par cœur, permettant ainsi à chaque tâche de traiter un nombre élevé de particules. La sauvegarde des données est assurée par un système de fichiers LUSTRE à 2 niveaux. C'est la quasi-totalité des nœuds fins qui a été utilisé dans le cadre de DEUS-FUR. Le suivi de l'évolution des 8192^3 particules a été effectué sur 38 016 tâches MPI distribuées sur 4752 nœuds avec 8 Gio par tâche pour un total d'environ 300 Tio de mémoire vive. La phase de simulation a été précédée par une étape de benchmark en configuration de weak-scaling qui s'est révélée être essentielle pour l'optimisation

1. Top 10 mondial au moment de la réalisation de la simulation

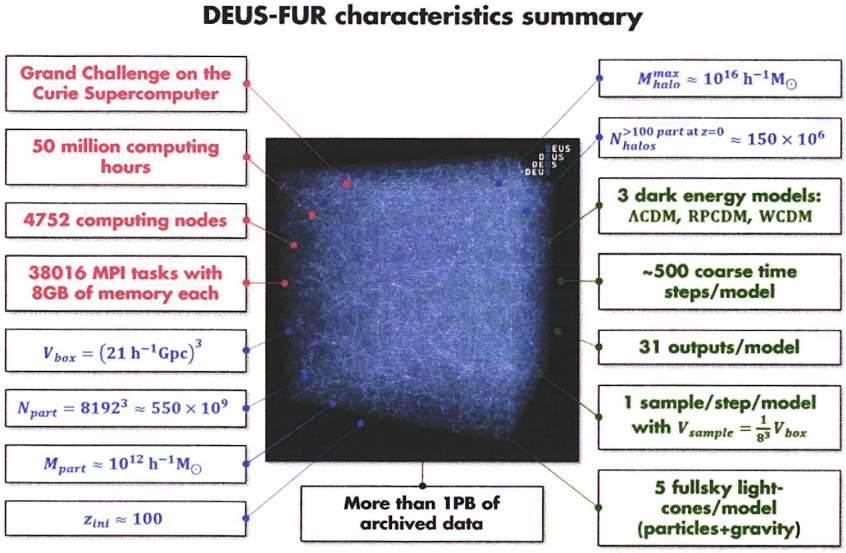


Figure 44 – Caractéristiques principales des simulations du Full Universe Run. Les principales propriétés numériques et physiques des simulations sont rappelées ici. Les paramètres cosmologiques correspondants ont été présentés au premier chapitre. Il s'agit de trois simulations 8192^3 de la taille de l'Univers Observable pour les modèles ΛCDM, WCDM et RPCDM basées sur les données de WMAP7 et réalisées sur le supercalculateur Curie. La participation à la réalisation de ces simulations a constitué un travail important dans le cadre de cette thèse. Les détails numériques de ce travail sont présentés dans deux articles fournis en annexe de ce manuscrit[36].

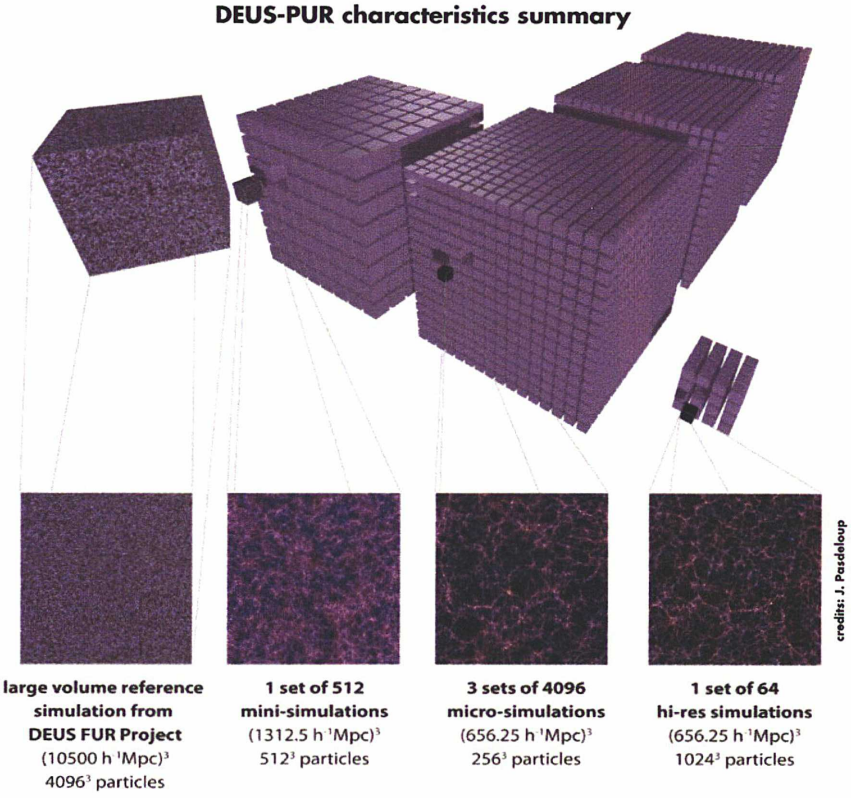


Figure 45 – Caractéristiques principales des simulations du Parallel Universe Run. Pour explorer l'hypothèse ergodique et les propriétés statistiques d'une grande quantité de simulations, le Parallel Universe Run a consisté à réaliser plus de 12 000 simulations ΛCDM dotées de phases différentes sur le supercalculateur Ada. Une simulation 4096^3 issue du Full Universe Run constitue la référence de cet ensemble de simulations dans le cadre de l'étude des effets de volumes finis. Davantage de détails sont donnés dans l'article fourni en annexe[119].

des différents aspects numériques présentés dans la suite. Les caractéristiques des configurations de weak-scaling sont résumées en figure 46.

5.3.2 Temps de calcul et communications

Le premier aspect numérique critique pour la réalisation des simulations a été le temps de calcul et la gestion des communications. Au total, 50 millions d'heures ont été consommées. L'évolution du temps de calcul et de communication comparativement au temps d'écriture tout au long d'une simulation est donnée en figure 48. Ainsi, ce temps passe d'environ 50 secondes par pas de temps au début de la simulation, lorsque l'Univers est encore peu structuré à environ 500 secondes à $z = 0$. Dans le même temps, le temps d'écriture des cônes, qu'il s'agisse des particules ou du champ gravitationnel décroît du fait de l'amoindrissement de la taille des coquilles à écrire. Comme on peut enfin le voir sur la figure 48, chaque simulation a été ponctuée de 31 phases d'écriture intense, allant de 1500 à 3200 secondes et permettant la sauvegarde de l'ensemble des données utiles grâce à une stratégie sophistiquée d'organisation des données.

La scalabilité du code dynamique pour les quatre configurations de weak-scaling 46 est donnée en figure 49. La scalabilité d'un code Particle-Mesh résolvant l'équation de Poisson par transformée de Fourier rapide sans raffinement est donnée comme référence, le code utilisé se basant, lui, sur un solveur multi-grille[121] permettant parfois d'atteindre de meilleures performances. Au tout début du run, sur 38 016 tâches, l'efficacité est de l'ordre de 60%, puis elle tombe à 55% lors du déclenchement du premier niveau de raffinement avant de remonter progressivement jusqu'à 75%. Au total, plus de la moitié du temps de simulation est passé dans une situation d'efficacité allant au delà de 65%.

Atteindre une telle scalabilité a demandé de revoir certains schémas de communication, notamment concernant les premiers niveaux de raffinement, à l'origine basé sur des communications aynchrones du type `MPI_Isend` et `MPI_Irecv`. Le passage à des communications synchrones `MPI_Send` et `MPI_Recv` ainsi qu'à de la diffusion de messages `MPI_Bcast` pour les premiers niveaux partagés par tous les processus a conduit à une amélioration significative des performances. Plusieurs tests ont cependant montré que ce genre d'optimisations étaient très dépendant des machines, des compilateurs et de la configuration MPI. La reproductibilité de telles méthodes d'optimisation ne peut donc être garantie ou même généralisée. Chaque simulation qui vise à utiliser l'entièreté d'une machine est unique et doit donc en ce sens faire l'objet de réglages qui lui sont propres.

5.3.3 Mémoire

Le second aspect numérique critique dans le cadre du Full Universe Run a été l'occupation mémoire. En effet, si Curie avait été choisie pour la réalisation de simulations à 8192^3 particules, c'est en outre à cause de la quantité élevée de mémoire dont elle dispose à la fois par cœur et au total. Pour être menée à bien, chaque simulation devait satisfaire un critère essentiel lié au matériel : que le besoin en mémoire ne dépasse pas 64 Gio par nœud, système et buffers MPI compris, et ce à aucun moment de la simulation. Le non-respect de cette condition n'aurait d'autre conséquence que d'entraîner un « swap » quelque part sur la machine, et donc au final, un blocage de la simulation. Pour maximiser la précision du calcul tout en évitant une telle situation, des tests préliminaires avaient été effectués sur les nœuds grossiers de la machine. Cela avait notamment permis de fixer le critère de déclenchement de raffinement à 14 particules par cellule. Le nombre total de cellules a ainsi pu être contrôlé sans affecter la précision sur deux observables : le spectre de puissance et la fonction de masse. En plus de ces tests préliminaires, des optimisations avaient déjà été effectuées pour adapter le code à 25 000 processus MPI sur une configuration Blue Gene P pour des phases précédentes du projet Dark Energy Universe Simulation. À cette occasion, l'occupation mémoire avait déjà été réduite d'un facteur ~ 3 par l'utilisation de tableaux de pointeurs alloués dynamiquement en lieu et place de tableaux dont la taille était proportionnelle au nombre de tâches MPI multiplié par le nombre de niveaux de raffinements.

Weak-scaling configurations of the Full Universe Run			
Particles	**Number of MPI tasks**	**Memory per MPI task**	**Number of nodes**
1024^3	74	8 GiB	10
2048^3	594	8 GiB	75
4096^3	4752	8 GiB	594
8192^3	38016	8 GiB	4752

Figure 46 – Liste des quatre configurations de weak-scaling utilisées. Les benchmarks ont joué un rôle essentiel dans la mise en place et l'optimisation des codes du Full Universe Run, ces quatre configurations permettant de mesurer la scalabilité des performances de 1024^3 particules à 8192^3 particules.

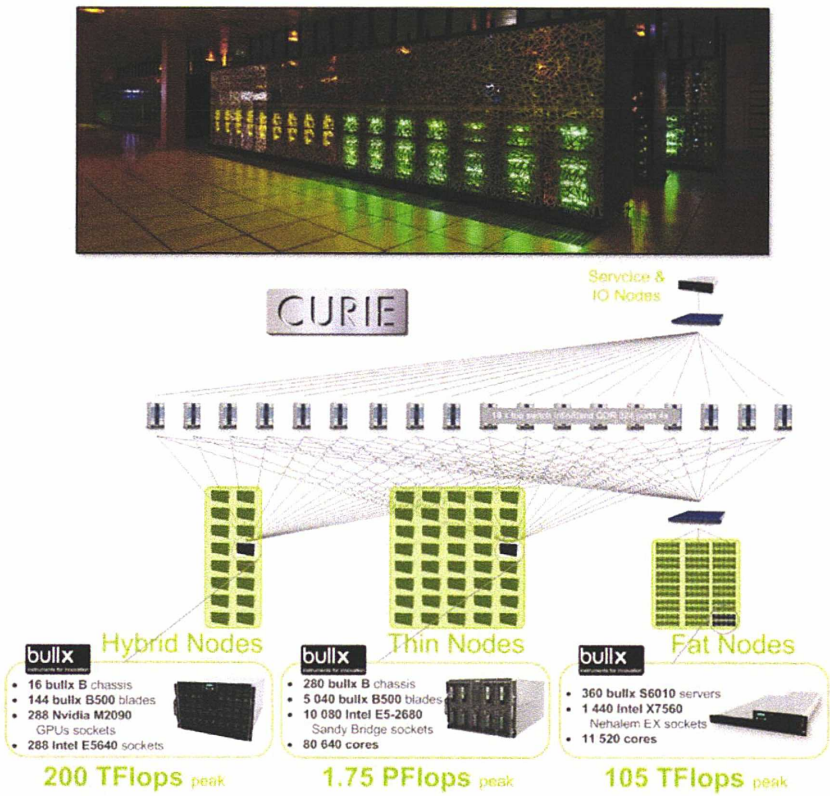


Figure 47 – Partitions du supercalculateur Curie. Le découpage de la machine en trois types de nœuds permet de résoudre différents types de problématiques numériques : les nœuds larges disposent d'une mémoire plus importante, les nœuds hybrides disposent de cartes accélératrices tandis que les nœuds fins regroupent la plupart de la capacité de calcul. C'est sur ces derniers qu'à été effectué le Full Universe Run.

OBSERVATOIRE DE PARIS

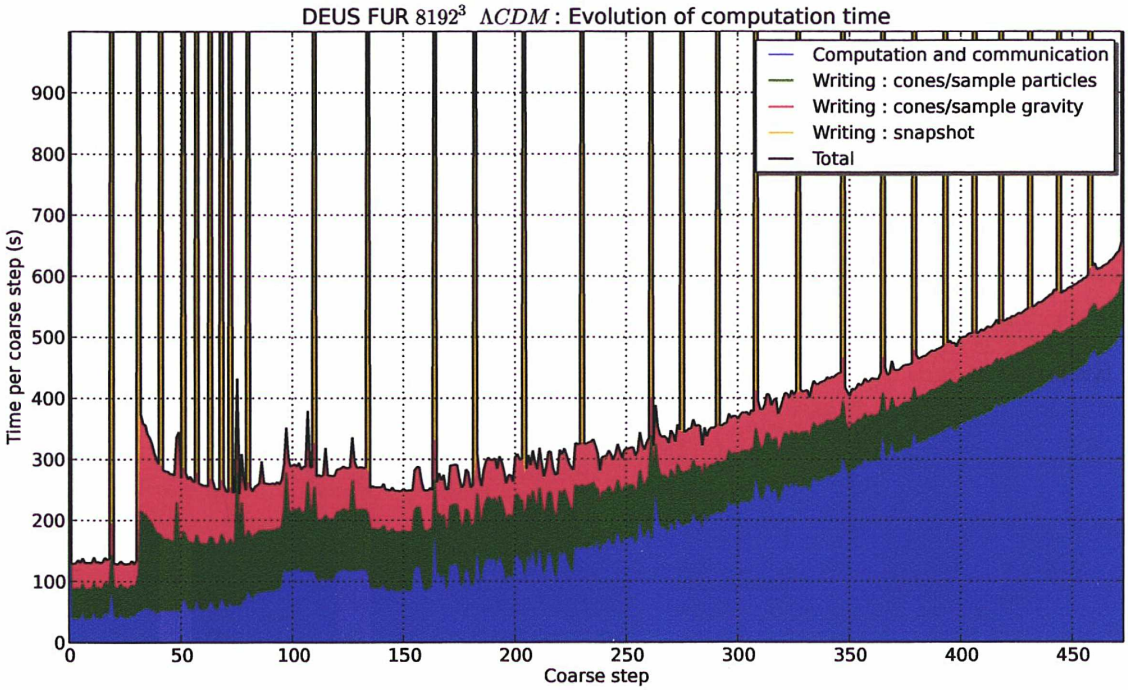


Figure 48 – Évolution du temps mis pour les différents aspects du calcul au cours d'une simulation. En bleu, le temps de calcul et de communication, en vert et en rouge, l'écriture des données des cônes et des sample à chaque pas de temps, et en orange l'écriture des snapshot principaux. Au total, 31 pas de temps sont écrits sur les disques au cours de chaque run. Peu à peu au cours d'une simulation, le temps consacré à la partie calcul pure s'accroît du fait de la structuration.

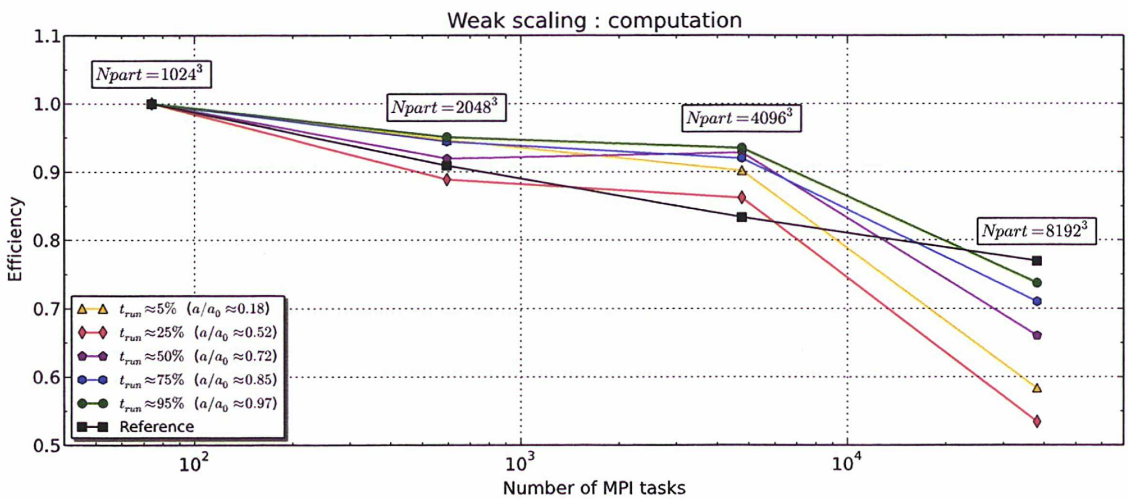


Figure 49 – Scalabilité du temps de calcul. La scalabilité du temps de calcul est ici évaluée par rapport à un cas de référence à 1024^3 particules en configuration de weak-scaling. Au début de la simulation, la scalabilité tombe à 55 % avant de remonter progressivement jusqu'aux alentours de 75 %. La référence, en noire, correspond au cas idéal d'un code PM-FFT.

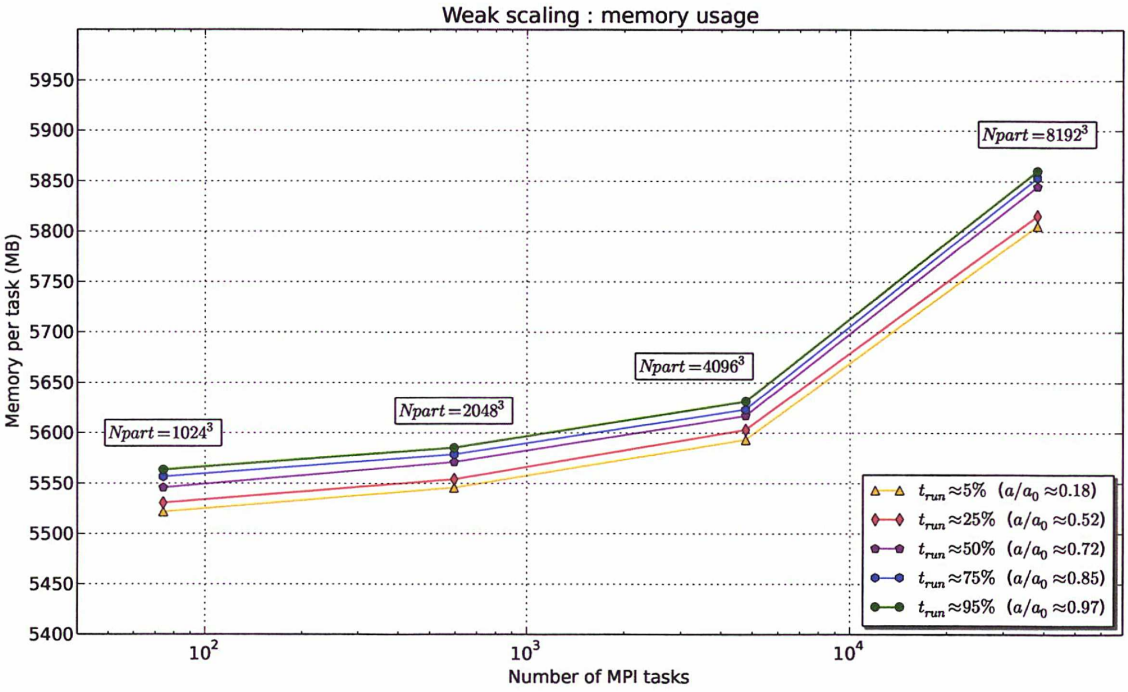


FIGURE 50 – Occupation mémoire. Ci-dessus sont illustrés les différents taux d'occupation mémoire au cours des simulations et pour des configurations en weak-scaling. Étant donné que la plupart des tableaux sont alloués au tout début des simulations, l'occupation mémoire évolue relativement peu. Elle est d'un peu plus 5.8 Gio par tâche MPI, sachant que la limite totale est de 8 Gio par tâche.

En configuration de weak-scaling, le passage de 74 à 38016 tâches entraîne un surcoût mémoire très limité, de l'ordre de 5%.

Un point qui s'est révélé être essentiel pour l'occupation mémoire pour le portage sur la machine CURIE a été le réglage des paramètres des buffers MPI. En effet, les phases de tests préliminaires avaient montré, via l'utilisation de VALGRIND que 9 tâches utilisaient jusqu'à 16 Gio de mémoire dans le cadre d'une configuration optimale du point de vue de la vitesse d'exécution. Le passage à une configuration à empreinte mémoire réduite, de même que le remplacement de communications point à point par des `MPI_Bcast` a permis de repasser à une situation d'utilisation de moins de 8 Gio par tâche, situation illustrée en figure 50. Sur cette figure, il apparait clairement que la très grande majorité de la mémoire est allouée au démarrage du code, la consommation totale augmentant peu d'un pas de temps à l'autre. Cela s'explique par le fait que la majorité des tableaux destinés à stocker les informations relatives aux particules et à la grille ne font l'objet d'aucune réallocation dynamique au cours d'un run : ils sont alloués une fois pour toute à chaque redémarrage de l'application. Lorsqu'ils arrivent à saturation, le code s'arrête, obligeant l'utilisateur à réévaluer la taille de ces tableaux. Cette gestion « manuelle » des erreurs d'allocation mémoire n'est pas limitante dans un contexte de simulation numérique haute performance et permet une meilleure gestion de la mémoire du point de vue utilisateur.

5.3.4 Entrées et sorties

Le dernier point en ce qui concerne l'utilisation du code numérique porte sur les entrées et sorties, qui ont été, là aussi, un véritable défi d'optimisation. Gérer 31 phases d'écriture de l'ordre de 100 Tio chacune par simulation n'est en effet pas tâche aisée. En plus de ces intenses phases d'écriture, un morceau de la simulation ainsi que les coquilles des cônes de lumière ont été extraits à chaque pas de temps, comme illustré sur la figure 48. Pour gérer de tels volumes de données, une phase de benchmarks de la machine destinée à déterminer les débits maximum d'écriture s'est révélée être nécessaire.

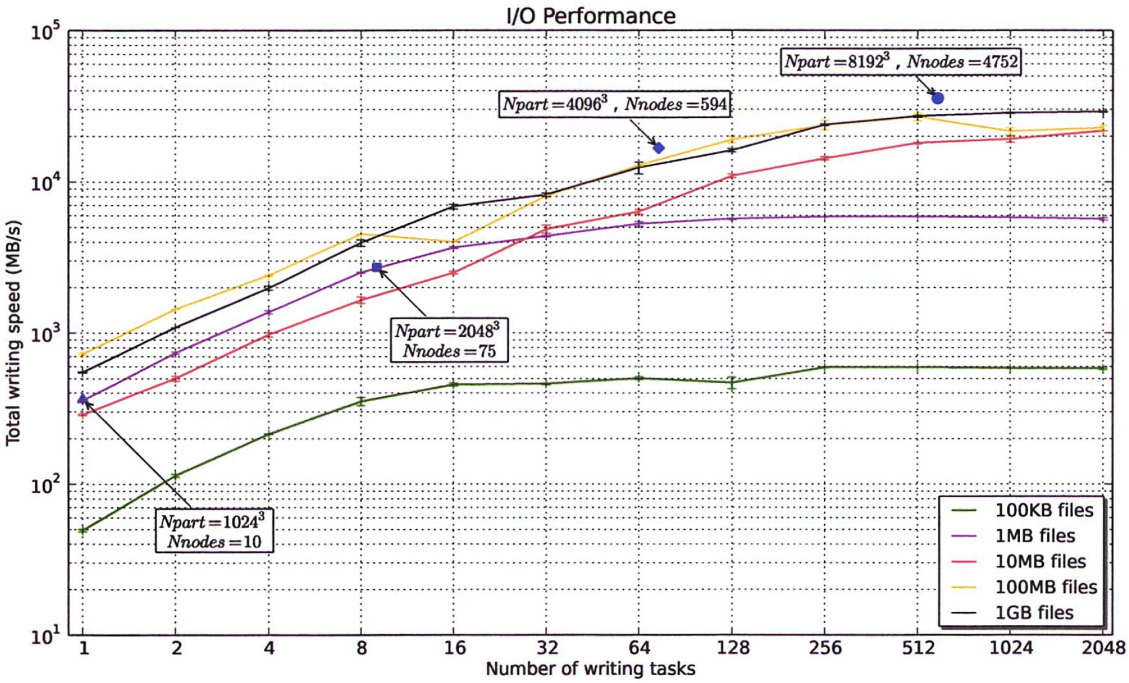


Figure 51 – Performance des entrées/sorties. Pour maximiser l'utilisation de la bande passante, des benchmarks ont été effectués sur la machine pour différentes tailles de fichiers et pour différents nombres de tâches. Au-delà de 512 tâches, les benchmarks révèlent que la vitesse globale d'écriture sature. Dans la pratique, nous avons fixé le nombre d'écritures simultanés à 594 .

Les résultats sont présentés en figure 51. La détermination du nombre optimal d'écritures simultanées était en effet un point important : un nombre inférieur au nombre optimal allait se traduire par une augmentation de la durée du calcul tandis qu'un nombre supérieur pouvait se traduire par une instabilité du système de fichiers. De telles instabilités ont notamment posé problème non à l'écriture, mais à la lecture des fichiers de paramètres initiaux. En effet, l'accès simultané par 38016 tâches à un fichier de quelques Kio suffisait originellement à faire tomber le système. Le problème a été résolu en attribuant à un seul processus la tâche de lire le fichier en question, puis en diffusant le contenu de celui-ci via des `MPI_Bcast`.

Les résultats des benchmarks révèlent que la vitesse maximale d'écriture globale peut être atteinte pour des fichiers de plus de 100 Mio : cette vitesse sature au-delà d'environ 512 tâches. Conformément à ces résultats, le code dynamique a été configuré par groupes de 64 tâches avec un système de tickets d'écriture au sein de chaque groupe, conduisant au total à 594 écritures simultanées, sachant que la taille typique des fichiers en jeu est de l'ordre du gigaoctet. Pour la gestion des cônes, dont le volume de données varie de pas de temps en pas de temps, c'est un ticket adaptatif qui a été implémenté, là encore pour maximiser l'utilisation de la bande passante. Dans tous les cas, le format de fichier qui a été adopté est un format binaire Fortran permettant d'atteindre les débits maximum. Rétrospectivement, c'est entre autres choses dans le cadre de la gestion des entrées et sorties parallèles de ce type que la parallélisation hybride du type MPI/OpenMP peut présenter certains avantages comparativement à une pure configuration MPI. En effet, cela simplifie grandement les phases d'écritures intense et réduit du même coup le nombre de fichiers à écrire : un thread par nœud peut se consacrer à l'écriture de toutes les données du nœud tandis que les autres threads continuent les calculs.

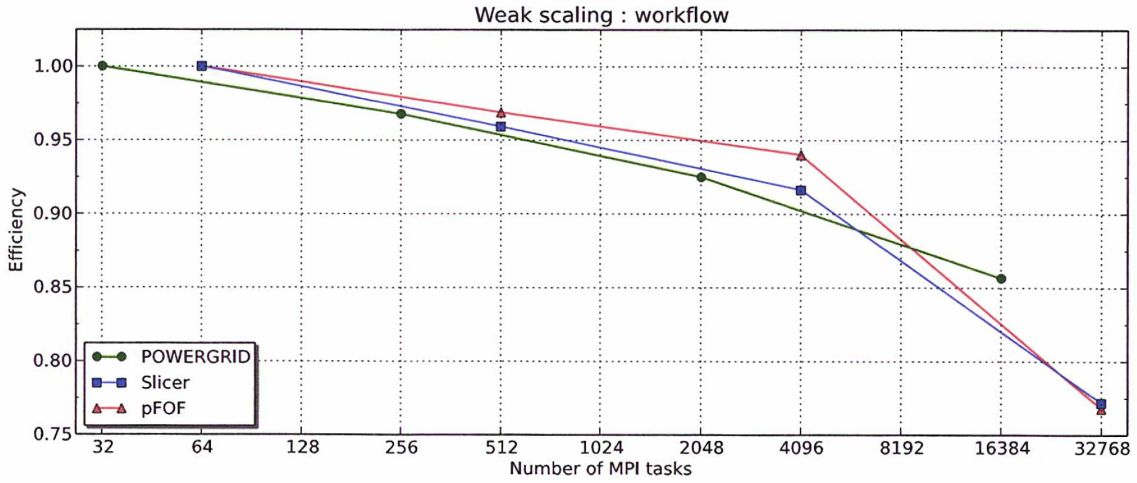


Figure 52 – Scalabilité des codes de post-traitement. Tous les aspects des simulations doivent être optimisés. Non seulement le code dynamique bien entendu, mais aussi les codes annexes dont la figure ci-dessus présente la scalabilité.

5.3.5 Pré et post-traitements

Enfin, le code dynamique ne fait pas tout : les phases de pré et de post-traitement sont tout aussi importantes et nécessitent autant d'attention. En ce qui concerne la génération des conditions initiales, le pré-découpage en domaines sur la courbe de Hilbert utilisée par le code dynamique pour la parallélisation était essentiel. Sans ce pré-découpage, cela aurait forcé le code dynamique a redistribué les données initiales juste après leur lecture et cela aurait pu conduire à une durée d'initialisation de l'ordre d'un ou deux jours à comparer à une vingtaine de minutes dans le cadre du pré-découpage. Au niveau des post-traitements, à l'ère du petascale, celui-ci ne peut plus consister en une phase unique. En d'autres termes, il est illusoire de chercher à analyser directement les données brutes à l'aide de langages de scripts. Il est nécessaire, du moins dans notre cas, d'introduire une étape de post-traitement intermédiaire permettant de passer des fichiers bruts à des quantités physiques pré-sélectionnées. Les codes chargés de ces analyses sont des codes lourds, dans le sens où ils doivent traiter les centaines de teraoctets générés par le code dynamique. La scalabilité des trois principaux codes utilisés est présentée en figure 52 : PowerGrid est le code chargé du calcul des spectres de puissance, PFOF est le code chargé de la détection des structures et le Slicer est le code chargé du redécoupage des données brutes en domaines cubiques facilement manipulables. Comme on peut le voir, ces codes sont eux-mêmes des codes orientés calcul haute performance nécessitant une bonne scalabilité jusqu'à 32 768 tâches. À l'issue de la phase de post-traitement, l'ensemble des produits finaux ont été directement écrits sur le système d'archivage, les copies gérées par le système étant bien trop lentes. Lorsque la taille des fichiers était trop petite, ces derniers ont été fusionnés via des codes conçus spécialement pour l'occasion, l'utilisation d'outils comme `tar` s'étant révélée inutilisable car trop lente.

5.4 Limitations de la cosmologie numérique

5.4.1 Des succès aux limitations

Comme nous l'avons vu dans la section précédente, la cosmologie numérique nous a menée en une quarantaine d'années de simulations d'un seule structure avec quelques centaines de particules à des simulations de la taille de l'Univers Observable avec plusieurs centaines de milliards de particules. Le fait d'être aujourd'hui capable de traiter les problématiques de structuration à de si grandes échelles marque un tournant pour la thématique : désormais il ne sera plus possible d'augmenter la taille de boîte. Aussi trois voies s'ouvrent à nous : celle qui consiste à se concentrer sur l'amélioration de la

résolution et de la statistique des simulations, celle qui consiste à intégrer davantage de phénomènes physiques, et celle qui consiste à simuler la structuration à partir de principes de plus en plus fondamentaux. Mais avant de se lancer à l'assaut de l'Exascale avec ces problématiques en tête, il pourrait être utile de tirer les leçons de l'expérience accumulée au cours de ces quarante dernières années. Car pour parvenir à des simulations pétaflopiques telles que le Full Universe Run, il a fallu pousser les codes actuels au maximum de leur capacité, que ce soit en termes de complexité ou en termes d'optimisation, comme nous venons de le voir. Dès lors, dans un contexte où d'une part le portage sur des architectures hétérogènes va demander une connaissance toujours plus fine du fonctionnement des machines et où d'autre part l'exploration d'une multitude de modèles physiques va demander la conception de codes toujours plus étendus, la complexification de ces derniers semble être un passage obligé sur la route de simulations exaflopiques. Aussi, les initiatives pour adapter les codes existants sur de nouvelles architectures, pour implémenter des nouveaux types d'optimisations spécifiques à certains matériels, pour utiliser de nouvelles méthodes de parallélisation, pour concevoir de nouveaux algorithmes, pour mettre en place de nouveaux codes d'analyse ou pour développer de nouveaux solveurs se multiplient. Mais ce n'est pas de cela dont il va s'agir dans la suite. Au lieu de chercher à complexifier l'existant, nous allons nous interroger sur le dénominateur commun à toutes ces pistes d'exploration : la conception logicielle, et voir en quoi elle peut être améliorée. Autrement dit, nous allons essayer de prendre le problème à sa racine et voir comment « tordre » les compilateurs pour que la physique puisse s'y exprimer de la manière la plus naturelle et efficace possible.

Pour se faire, la suite du présent chapitre est divisé en trois parties. La première traite des limitations actuelles de la cosmologie numérique. Il y est question des leçons du Full Universe Run et de leurs implications plus larges sur les codes de simulation. La seconde propose des pistes de réflexion et les techniques sur lesquelles elles s'appuient pour tenter de résoudre les problèmes soulevés dans la première partie. Il y est question de métaprogrammation, de langages embarqués, de compilation et de bibliothèques actives. La troisième, enfin, prend du recul par rapport à ces techniques en vue de proposer un traitement à la fois générique et performant des problématiques qui ont trait à la cosmologie numérique, qu'il s'agisse de simulations dynamiques proprement dites ou d'analyse de données. Cette dernière partie prend entre autres appui sur certaines expérimentations qui ont pu être faites lors de l'implémentation d'une bibliothèque numérique, MAGRATHEA, autour de ces questions.

5.4.2 Leçons du Full Universe Run

Nécessité d'un « workflow »

La réalisation de simulations de 550 milliards de particules sur la quasi-totalité d'une machine pétaflopique nous a permis de tirer un certain nombre de conclusions sur le plan numérique. Tout d'abord, la gestion d'une chaîne de codes complexe partant de la génération des conditions initiales jusqu'au post-traitement ne peut plus faire l'objet d'un traitement « manuel », lent, complexe, et source d'erreurs. En lieu et place d'une telle gestion, l'utilisation de scripts d'automatisation semble de plus en plus adaptée, y compris pour le contrôle de la soumission des travaux, comme cela a pu être le cas pour le Parallel Universe Run. Une autre approche du problème, complémentaire, vise à l'intégration de plus en plus de composants dans les codes de simulation en vue de réaliser un maximum d'analyses à la volée, mais cela entraîne, de fait, une complexification de ces derniers.

Plusieurs types d'optimisation

Le second point concerne la stratégie d'optimisation. Lorsqu'un code est porté sur une machine, quatre aspects principaux peuvent être distingués :

- le temps de calcul
- les communications

- l'occupation mémoire
- les entrées et sorties

Souvent, tous ces aspects ne sont pas critiques simultanément si bien que l'on peut procéder à des optimisations basées sur des échanges entre catégories, comme illustré en figure 53 : on parlera dans la suite de « trading-based optimizations ». Par exemple, la résolution d'un problème de nature combinatoire peut demander un temps de calcul très important et une consommation mémoire très faible. Dans ce cas, il est possible, dans une certaine mesure, d'optimiser le temps de calcul en stockant un maximum de résultats intermédiaires : la réduction du temps de calcul s'accompagne ainsi d'une augmentation de l'occupation mémoire. Le caractère critique d'un aspect est relatif à une architecture donnée, cette dernière venant fixer limites de chacun d'entre eux, comme par exemple la mémoire maximale disponible. Lorsque trois aspects sur quatre sont critiques, il est encore possible de s'appuyer sur le quatrième pour optimiser les trois autres. Mais lorsque les quatre sont critiques simultanément, alors procéder à des « trading-based optimizations » n'est plus possible. Et cela était justement le cas du Full Universe Run relativement aux caractéristiques de la machine Curie. La seule solution qui demeure alors est d'optimiser chaque aspect indépendamment les uns des autres, ce qui s'avère être beaucoup plus difficile. Il peut s'agir de concevoir de nouveaux algorithmes, d'utiliser d'autres structures de données, de changer les schémas de communication, ou encore de modifier les formats d'entrée et de sortie. Mais plus on se concentre sur ce type d'optimisation, plus celles-ci deviennent spécifiques à l'architecture considérée. Autrement dit, on perd en généricité de l'application et celle-ci voit naître une nouvelle version spécialement conçue pour une machine ou une gamme de machines particulière. C'est par exemple le cas lorsqu'il s'agit d'utiliser des jeux d'instructions d'une génération de processeurs donnée pour accélérer certains types de calcul. Là encore, cela contribue à la complexification des applications comme souligné dans l'article présentant les leçons tirées du Full Universe Run fourni en annexe.

Sélection des données

Troisièmement, le stockage de l'intégralité des données n'est plus possible. De la même manière que des expériences comme celles menées au CERN ou des projets observationnels comme SKA filtrent ou projettent de filtrer en amont les résultats bruts pour ne garder que les évènements pertinents, une sélection des données au cours des simulations doit être effectuée. Il s'agit de trouver un équilibre entre ne garder que le strict minimum pour les analyses prévues au départ et en garder un maximum pour réaliser par la suite toutes les analyses possibles. Dans le cas du Full Universe Run, tout archiver à tous les pas de temps aurait nécessité plus de 100 Po d'espace disque. Cela n'étant pas techniquement possible, des redshifts d'intérêt ont été sélectionnés *a priori*, les données ont été post-traitées au fur et à mesure, puis elles ont été compressées et archivées.

Modification des codes existants

Enfin, la quatrième leçon importante, mais qui n'est pas spécifique au Full Universe Run, est que modifier les codes existants pour résoudre d'autres problèmes de ce pourquoi ils ont été prévus au départ peut s'avérer très complexe, même si les structures de données et les schémas numériques sont les mêmes. La question s'est par exemple posée pour gérer la propagation de la lumière au sein des cônes reconstruits à partir des simulations. Cela demande entre autres d'intégrer les équations géodésiques sur une grille de maillage adaptatif, grille justement implémentée dans le code dynamique. Dès lors deux voies sont possibles : modifier en profondeur le code dynamique en question, ou redévelopper un code spécifique pour cette tâche. C'est la seconde option qui a été choisie, mais en essayant de maximiser la généricité et les performances pour éviter certaines limitations des codes actuels, et en inscrivant cette approche dans une réflexion plus large sur la manière de construire les codes de simulation.

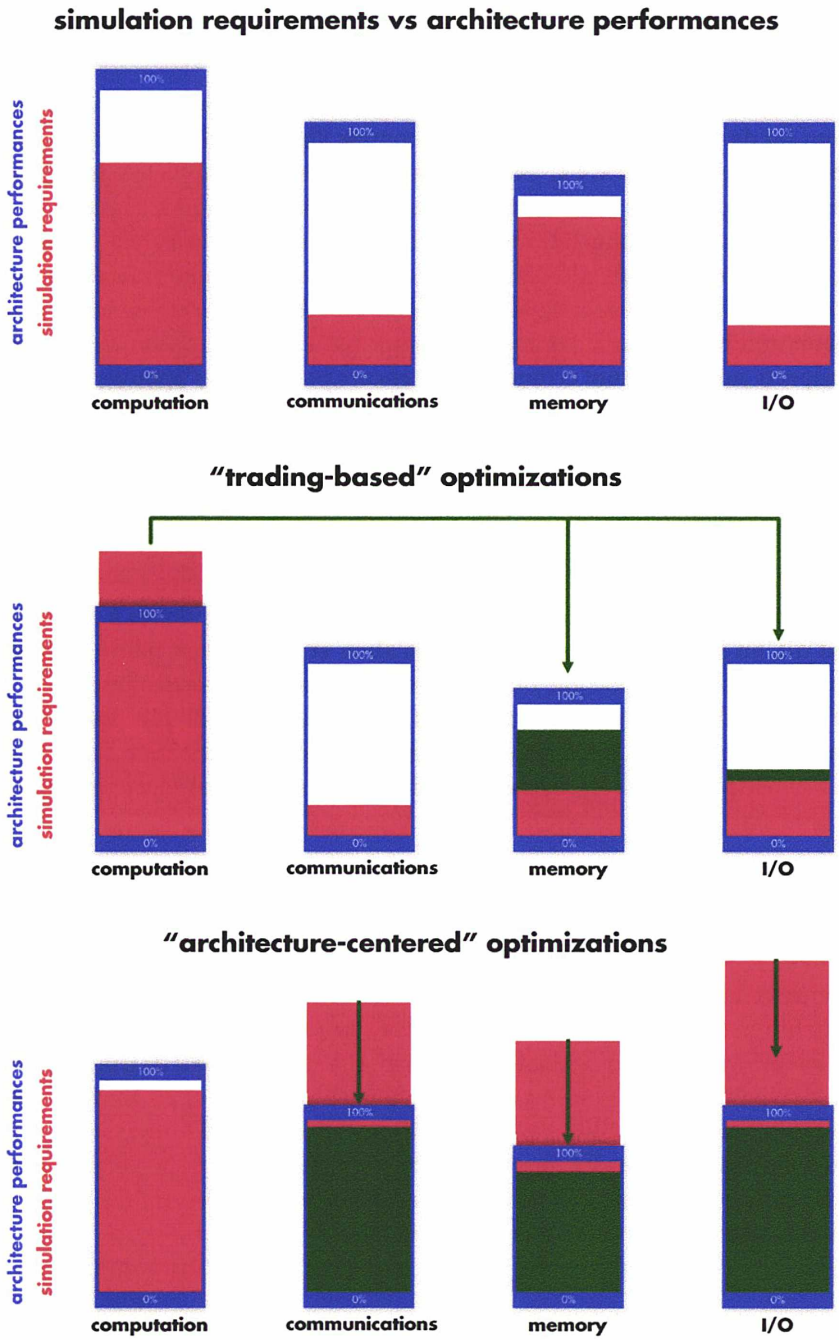


FIGURE 53 – Optimisations lors du portage d'un code sur une machine. Étant donnée l'architecture d'un supercalculateur et étant données les ressources nécessaires à la réalisation d'une simulation aux caractéristiques préalablement établies il peut s'avérer nécessaire de procéder à des optimisations sur les quatre composantes principales du calcul numérique : le temps de calcul, les communications, l'empreinte mémoire et les entrées/sorties. Lorsque les performances offertes par le supercalculateur dépassent les besoins d'une simulation, il n'est pas obligatoire d'optimiser le code lors du portage, comme illustré dans la partie supérieure de la figure. Lorsqu'un ou plusieurs aspects nécessaires à la réalisation de la simulation dépassent les moyens disponibles, il est possible de procéder à des « trading-based optimizations », c'est-à-dire des optimisations basées sur des transferts entre catégories. Il peut s'agir par exemple d'alléger la composante des performances pures par le stockage de variables temporaires dans la mémoire. Enfin, lorsque tous les aspects dépassent originellement les performances pouvant être atteintes sur le supercalculateur, il est nécessaire de procéder à des optimisations aspect par aspect, souvent dépendantes de l'architecture, comme illustré dans la partie inférieure dans la figure. Le Full Universe Run s'inscrivait dans cette dernière catégorie.

5.4.3 À la source de tous les maux : l'intrication

Intrication des différents aspects

Pour paraphraser Donald Knuth, l'on pourrait dire « intertwining is the root of all evil ». En effet, une partie significative des limitations actuelles provient d'une unique raison : l'intrication. C'est ce que l'on va tenter de montrer dans cette section. En général, trois parties peuvent être distinguées dans un code de simulation :

- ce qui est lié à la parallélisation : schémas de communication, sérialisation des données...
- ce qui est lié aux méthodes numériques : maillage, solveurs, intégrateurs...
- ce qui est lié à la physique : équations à résoudre, types de données...

Dans le cadre de simulations physiques, l'on pourrait naïvement penser que la troisième partie est ce qui constitue la majeure partie des codes et ce qui concentre l'attention de tous les développements. Dans la pratique, il s'agit en général de quelques lignes dispersées au sein de codes qui peuvent en comprendre plusieurs dizaines ou centaines de milliers. Le problème est que ces lignes sont en général imbriquées dans celles consacrées aux méthodes numériques, ces dernières étant elles-mêmes imbriquées au sein des schémas de parallélisation. En conséquence, il existe une forte interdépendance entre les différentes parties. C'est de là qu'est originaire la notion d'intrication. La première conséquence directe de cette intrication est qu'il souvent impossible de modifier uniquement la physique d'un code, même si les méthodes numériques et la parallélisation restent les mêmes. Ajouter des variables, ou modifier les équations implique généralement une refonte d'une partie substantielle de l'application concernée. Le physicien qui souhaite résoudre un problème légèrement différent se retrouve ainsi d'une part à devoir maîtriser en profondeur les méthodes numériques et les schémas de communication utilisés et d'autre part à écrire des dizaines ou des centaines de fois plus de lignes sur ces aspects que sur l'aspect purement physique. De façon très pratique, cela peut conduire à l'utilisation d'antipattern ou à la duplication excessive de code, qui constituent, à terme, des dettes techniques.

Gestion de version et non-commutativité des changements

Le manque de modularité lié à cette intrication entraîne souvent une augmentation substantielle du nombre de versions des codes les plus utilisés par la communauté, chaque groupe de recherche développant sa propre version répondant aux besoins de l'architecture visée et du problème physique étudié. Si ces changements pouvaient profiter à la communauté par l'intégration à la version « officielle » au bout d'un certain temps, cela ne poserait pas de problème. Mais l'intrication entraîne bien souvent mécaniquement la non-commutativité des modifications. Ainsi, si à partir d'un code original, un groupe de recherche décide d'implémenter une nouvelle physique puis se penche sur l'optimisation de l'application pour une plateforme particulière pendant qu'un autre groupe travaille à l'optimisation du code original pour une autre architecture, il est peu probable que ce dernier puisse profiter simplement de la nouvelle physique implémentée par le premier groupe. Pour cette raison, les versions spécialisées se multiplient et sont difficilement intégrées à la version officielle qui poursuit pendant ce temps son propre développement.

La question de l'expertise

L'intrication pose également la question de l'expertise. En effet, comme il n'est pas possible de modifier uniquement l'un des trois aspects identifiés précédemment, l'on se retrouve dans une situation où tous les intervenants doivent maîtriser l'ensemble des composants du code avant d'être en mesure d'apporter des modifications. Pire, cela peut empêcher les spécialistes d'une thématique particulière d'intervenir facilement sur l'application, et donc, réciproquement, cela peut empêcher la communauté de bénéficier de certaines expertises sur des points particuliers. Dans un contexte où les architectures

risquent de devenir de plus en plus hétérogènes avec à la fois des processeurs multicœurs, des accélérateurs graphiques et des cartes « manycores », c'est un problème central pour bénéficier des algorithmes les plus performants et des meilleures optimisations originaires de différentes communautés.

Explosion combinatoire

Mais l'une des conséquences principales de l'intrication a à voir avec l'augmentation de la complexité des codes. Et il s'agit là d'une problématique majeure lorsque l'on cherche à augmenter le nombre d'architectures ciblées, le nombre de problèmes physiques traités et le nombre d'algorithmes de pré ou post-traitement intégrés aux codes. Une façon intéressante de traiter cette question consiste à s'interroger sur la scalabilité du nombre de lignes de codes, abrévié SLOC [2] dans la suite, en fonction du nombre d'architectures N_A, du nombre d'algorithmes et de méthodes numériques N_M et du nombre de types de données N_D, souvent lié à la physique traitée. Autrement dit, au lieu de s'interroger sur la complexité d'exécution d'un algorithme, on s'interroge ici sur la complexité des codes de simulation mesurée à l'aune du nombre de lignes de codes dont ils sont composés. Et lorsque les différents aspects sont intriqués et que l'on souhaite bénéficier des meilleures performances possibles comme c'est le cas dans le contexte du calcul haute performance, on a quelque une scalabilité qui peut tendre vers :

$$\boxed{\text{SLOC} = \mathcal{O}\left(N_D \times N_M \times N_A\right)} \tag{5.1}$$

C'est-à-dire que pour bénéficier des meilleures performances, il faut en général concevoir une version de chaque fonction spécifique à chaque combinaison de type de données, d'algorithme et d'architecture. Ainsi, de façon très qualitative, un programme pour lequel :

- $N_D = 2$ avec des quantités scalaires et vectorielles
- $N_M = 3$ avec des algorithmes pour calculer des moyennes, des sommes et des normes
- $N_A = 3$ avec une version séquentielle, une version optimisée pour les architectures à mémoire partagée, et une version optimisée pour les architectures à mémoire distribuée

demandera environ $2 \times 3 \times 3 = 18$ fois plus de lignes de codes qu'une seule des combinaisons. Bien entendu, des optimisations sont toujours possibles par factorisation de code, et il ne s'agit là que de dessiner des tendances générales. Mais dans l'ensemble, une approche classique du problème conduit à une explosion combinatoire du nombre de lignes de code nécessaire pour traiter tous les cas avec des performances maximales. En supposant que la contribution de chaque développeur intervenant sur le projet dépend linéairement du temps, deux solutions principales sont possibles :

- attirer un nombre de contributeurs de plus en plus grand pour traiter l'explosion combinatoire
- ne traiter que les cas d'utilisation nécessaires à chaque équipe de recherche

D'un point de vue pratique, c'est naturellement le second cas qui se produit, ce qui conduit aux problématiques discutées précédemment comme la création d'un grand nombre de versions spécialisées incompatibles entre elles.

5.4.4 Problématique

Traitement de l'explosion combinatoire

La description du problème en termes d'explosion combinatoire permet de formuler très simplement les caractéristiques du type de solution attendue. En effet, de manière analogue à ce que l'on cherche à faire dans le cadre d'une optimisation algorithmique, on cherche une façon de passer de 5.1 à :

$$\boxed{\text{SLOC} = \mathcal{O}\left(N_D + N_M + N_A\right)} \tag{5.2}$$

c'est-à-dire une situation idéale où le nombre de lignes de codes dépend de la somme et non plus du produit du nombre de types de données, du nombre d'algorithmes et du nombre d'architectures.

2. Source lines of code.

Rôle d'un compilateur

Or, il s'agit précisément du type de travail pouvant être accompli par un compilateur : assembler le puzzle formé par des morceaux de code indépendants et générer en sortie l'exécutable le plus performant possible. Si de telles méthodes peuvent être couramment employées dans le domaine de la recherche en informatique, elles sont relativement peu répandues pour le développement de simulations physiques. Une telle approche permet de séparer et d'isoler les problèmes qui ont trait à la physique, ceux qui ont trait aux méthodes numériques et ceux qui ont trait aux architectures et à la parallélisation. Chaque composant peut alors être développé indépendamment, le compilateur se chargeant à la fin d'assembler toutes les parties et d'optimiser le résultat pour le supercalculateur ciblé. En apportant une réponse au problème de l'intrication, cette solution résout du même coup les trois autres problèmes évoqué dans la section précédente. D'abord, elle vient rétablir la commutativité des modifications du code. En effet, chaque équipe de recherche peut développer sa version d'un algorithme, ou son optimisation particulière sans avoir à intervenir sur les autres parties. La modularité du code est ainsi assurée. De plus, le projet peut bénéficier de multiples expertises puisqu'il n'est plus nécessaire d'en maîtriser tous les aspects dans leurs moindres détails pour implémenter des modifications. Enfin, la complexité est considérablement réduite, puisque c'est au compilateur qu'il revient de gérer l'explosion combinatoire décrite au paragraphe précédent et non plus aux développeurs, ces derniers pouvant se concentrer sur des problèmes rendus les plus indépendants possibles. D'un point de vue de l'exploration des problématiques physiques le gain est considérable, puisque l'on peut traiter des problèmes très différents sans avoir à intervenir sur la parallélisation ou les maillages si ces derniers ont été pensés au départ d'une façon suffisamment générique.

Compilation et langages dédiés

Si l'on prend le problème à sa racine, la solution réside donc dans le fait de concevoir un compilateur, et donc en fait un langage, dédié à la cosmologie. Dans cette optique, trois solutions principales existent :

- la création d'un DSL, « Domain Specific Language » et du compilateur qui l'accompagne
- la création d'un précompilateur ou d'un traducteur
- la création d'un EDSL, « Embedded Domain Specific Language », c'est-à-dire d'un langage dédié embarqué dans un autre langage

La première, celle des langages dédiés, attaque frontalement le problème par la conception d'un nouveau langage spécifique à la thématique dans le cadre de laquelle il s'inscrit. Les DSL sont construits en opposition aux langages généralistes, abréviés GPL pour « General Purpose Language » tels que le C, le C++, Python, Java... Un bon exemple de langage dédié est LaTeX, en l'occurrence pour la composition de documents : tout ce qu'il est possible de faire avec LaTeXpourrait être accompli dans un langage généraliste comme le C, mais le fait de se restreindre à un domaine d'application particulier permet de proposer une syntaxe et une grammaire beaucoup plus « naturelle » pour traiter les problématiques du domaine en question. Ainsi, concevoir un DSL pour la cosmologie permettrait de maximiser l'expressivité de la physique tout en réduisant le recours à des centaines de lignes consacrées au maillage ou à la parallélisation lorsque l'objet principal n'est pas là. Mais cette approche s'accompagne de plusieurs inconvénients. Tout d'abord la conception d'un langage et d'un compilateur à part entière n'est pas une tâche aisée et demande un investissement initial non négligeable de la part de personnes ayant développé une expertise informatique conjointement à l'expertise du domaine d'application visé. Ensuite, cela requiert des utilisateurs l'apprentissage d'un nouveau langage de programmation, et peut empêcher du même coup la compréhension des codes sources pour les personnes extérieures au domaine d'application. De plus, le fait que la communauté soit réduite implique un support restreint et des capacités d'optimisation moindres de la part du compilateur dédié, ce qui n'est pas sans poser des problèmes dans le cadre du calcul haute performance.

La seconde approche, plus simple à mettre en place, consiste à créer un précompilateur, c'est-à-dire un utilitaire capable de traduire toute ou partie de code dans un autre langage. Cela permet par exemple d'étendre un langage généraliste avec des instructions spécifiques au domaine d'application. MPI, pour la parallélisation, ou Qt pour la création d'interfaces graphiques illustrent ce concept. Les utilisateurs ont toujours besoin d'apprendre un nouveau langage, mais cette fois beaucoup plus limité, et la compréhension est facilitée pour les non-experts du domaine. L'optimisation de l'exécutable étant à la charge du langage cible, le problème du caractère restreint de la communauté ne se pose pas. En revanche, cette approche alourdit le processus de compilation. Cette approche à base de précompilateurs, de préprocesseurs, ou de traducteurs est déjà déployée dans le cadre de bibliothèques scientifiques relativement importantes, comme par exemple le code Cactus[122], très utilisé dans la communauté relativiste. Cependant, force est de constater que la complexification de la compilation et la multiplication des utilitaires peut constituer une barrière à l'entrée pour les nouveaux utilisateurs.

Reste alors la troisième approche, celle des langages dédiés embarqués, les EDSL, qui permet de combiner les avantages des deux solutions précédentes. Au lieu de chercher à concevoir un tout nouveau langage, il s'agit d'utiliser les propriétés d'un langage existant pour embarquer au sein de celui-ci les éléments constitutifs d'un DSL. Dans ce cas, on règle le problème de la compilation puisque celle-ci est totalement prise en charge par le langage hôte. D'autre part, les utilisateurs qu'ils soient experts du domaine ou non, gardent l'impression de pouvoir exprimer les problématiques du domaine ciblé de façon plus naturelle au sein d'un langage et d'un environnement connu. En revanche, comme il s'agit en général d'intervenir dans le processus de compilation lui-même, l'implémentation peut requérir une maîtrise de ce processus : si l'utilisation d'un EDSL comporte peu de difficultés et de barrières à l'entrée, sa conception peut être complexe. Dans la pratique, un EDSL peut prendre la forme d'une bibliothèque active : c'est-à-dire une bibliothèque capable de rétroagir sur le code au sein duquel elle est utilisée au moment de la compilation. Eigen[123] est un bon exemple d'une telle bibliothèque pour l'algèbre linéaire : du point de vue utilisateur, il s'agit de manipuler une bibliothèque à l'apparence tout à fait standard qui permet une très bonne expressivité des problématiques d'algèbre linéaire, mais lors de la compilation la bibliothèque rétroagit sur le code, en l'occurrence via de l'« expression template »[124] pour apporter toutes les modifications nécessaires à l'optimisation du calcul matriciel. Dans ce cas, toutes les difficultés sont reportées du côté des concepteurs et non du côté utilisateur, pour qui les inconvénients majeurs des deux approches précédentes sont résolus.

Problématique et objectif

Pour résoudre le problème de l'intrication, ou du moins pour le réduire autant que faire se peut, pour contourner celui de l'explosion combinatoire, et pour dépasser certaines limites actuelles de la cosmologie numérique dans le contexte de l'Exascale, nous allons donc réfléchir à la conception d'une librairie active en vue d'implémenter un EDSL pour la cosmologie. Le but n'est pas, dans un premier temps, de fournir un EDSL complet, mais il est plutôt d'en fournir une preuve de concept et de dessiner les contours d'une architecture logicielle suffisamment générique pour traiter la plupart des problématiques abordées dans le cadre des simulations cosmologiques. La prochaine section est consacrée aux solutions techniques disponibles pour accomplir cet objectif tandis que la suivante se concentre sur la conception de l'architecture logicielle sus-citée.

5.5 Programmation générique, programmation générative et métaprogrammation

5.5.1 Choix techniques

Dans un contexte où l'on cherche à concevoir un langage dédié embarqué pour la cosmologie, le choix du langage hôte est important. Celui-ci doit répondre à plusieurs critères. D'une part, il doit ré-

pondre à un critère de performances, puisqu'elles sont ici au cœur du problème : on peut donc exclure les langages de script (BASH...) et les langages interprétés (PYTHON...). Il est aussi nécessaire d'avoir un contrôle très bas niveau pour mettre en œuvre certaines optimisations : on peut exclure les langages managés (C#...) et les langages comme JAVA. Reste les langages compilés comme FORTRAN ou C. D'autre part, il doit laisser un accès suffisant au processus de compilation pour permettre l'implémentation d'un EDSL. Reste alors principalement le C++ et le D. Notre choix s'arrêtera sur le premier pour des critères d'interfaçage avec le C et pour des critères de taille de communauté. Cette dernière est fondamentale pour assurer des performances maximales aux compilateurs et pour assurer la maintenabilité à long terme qui est critique pour des codes scientifiques parfois destinés à être utilisés pendant une voire plusieurs décennies. Les possibilités techniques offertes par le D sont néanmoins très prometteuses et tout ce qui sera discuté dans la suite pourrait faire l'objet d'une adaptation dans ce langage.

Nous ferons également le choix délibéré de ne pas s'appuyer sur des bibliothèques tierces. Comme il s'agit d'explorer les pistes offertes par le langage nous chercherons en effet à maintenir une compatibilité maximale avec les compilateurs et toutes les architectures en minimisant le nombre de dépendances. À terme, il serait bien entendu possible, et même souhaitable de bénéficier de l'interfaçage avec des bibliothèques tierces, mais dans un premier temps, on se limitera au langage et à la bibliothèque standard. On utilisera toutefois la dernière version qui facilite grandement certains aspects de la création de bibliothèques actives. En d'autres termes, on s'appuiera sur le standard ISO/IEC 14882:2011[125]. Deux projets de normalisation sont actuellement en cours pour inclure d'une part des concepts[126] et d'autre part des outils de réflexion[127], d'introspection et de réification à la compilation dans le langage. Il s'agit de projets à long terme qui devraient conduire au final à un meilleur support des techniques présentées dans la suite.

5.5.2 Programmation générique et factorisation de code

Position du problème

L'une des clefs du traitement de l'explosion combinatoire est la factorisation de code via la programmation générique. L'objectif est de n'écrire qu'une seule fois des fonctions qui peuvent opérer sur plusieurs types de variables. Le problème que l'on cherche à résoudre peut être posé au moyen de l'exemple suivant : comment n'écrire qu'une seule fonction `swap` destinée à échanger les valeurs de deux variables indépendamment de leur type ? En C ou en FORTRAN90 [3] une approche basique consiste à expliciter tous les cas possibles : `swap_int`, `swap_char`, `swap_double`..., le contenu de la fonction pouvant être éventuellement factorisé à l'aide de macros. La spécification des fonctions mathématiques fournies dans `<math.h>` est bâtie sur cette approche. Cela conduit évidemment à de la redondance de code, et ne permet pas de traiter tous les cas puisque l'utilisateur est toujours libre de créer ses propres structures de données pour lesquelles aucune fonction n'aura été spécifiée. L'utilisation de `typedef` en C ou de `kind` paramétrés en FORTRAN ne solutionne pas le problème, puisqu'au final une seule fonction est générée même si le type traité peut être modifié plus facilement à la compilation. La solution la plus générale est de passer par des pointeurs génériques `void*` ou de la réinterprétation de type via des `transfer` FORTRAN. Mais, ce faisant, on perd toutes les propriétés de typage fort en opérant directement au niveau de la représentation binaire. Existe-t-il d'autres solutions qui préservent toutes la logique liée aux types ?

Programmation orientée objet

C'est là que peut intervenir la programmation orientée objet, qui a, force est de constater, encore peu pénétrer le monde de la cosmologie numérique. Au lieu de spécifier une fonction pour tous les

3. Dans toute la suite on fera référence à ces deux langages car ils sont actuellement les plus utilisés dans le domaine de la cosmologie numérique.

types possibles, on implémente une fonction pour chaque classe de base [4] `swap(Base& x, Base& y)` et on laisse l'upcasting [5] se charger des conversions de type. Ainsi, chaque classe dérivée de la classe de base du côté utilisateur pourra faire usage de la même fonction via le polymorphisme. Cette notion de polymorphisme s'accompagne de la notion de virtualité qui permet au programme de faire appel à la bonne version de chaque fonction en se déplaçant dans le graphe d'héritage. L'orienté objet est très utile pour structurer un programme en concepts, en particulier lorsqu'il s'agit de projets de taille conséquente où la notion d'architecture logicielle devient importante. L'héritage permet en effet de raffiner les types en allant vers des spécifications de plus en plus précises. Mais si elle permet de structurer les codes d'une façon plus élégante, la programmation objet comporte un défaut majeur dans un contexte HPC : l'impact non négligeable sur les performances des programmes.

En effet, toutes ces notions d'héritage et de graphe d'appel des fonctions sont traitées à l'exécution. Cela introduit d'une part un coût en mémoire et un coût en temps de calcul. En mémoire tout d'abord, puisque la virtualité est généralement implémentée en termes de `vtable` et de `vptr` . La première est spécifique à chaque classe, commune à toutes les instances, et permet de lister les versions des fonctions virtuelles propres à la classe en question, tandis que les seconds permettent à chaque instance d'utiliser la bonne table dans la hiérarchie de classes. D'un point de vue de la mémoire, cela entraîne un surcoût mutualisé pour chaque classe proportionnel au nombre de fonctions virtuelles et ajoute un pointeur pour chaque instance quel que soit le nombre de fonctions virtuelles tant que celui-ci est non-nul. Les conséquences pratiques peuvent être assez inattendues. Ainsi, si une particule est repérée par trois flottants :

```
1 struct particle {float x, y, z;};
```

et qu'une simulation en contient 550 milliards, cela se traduit par une empreinte mémoire de $3 \times 4 \times 8192^3 = 6$ Tio. Le simple ajout du mot-clef `virtual` devant le destructeur de la classe `particle` tel que cela pourrait être conseillé par la plupart des ouvrages sur les bonnes pratiques à adopter dans le cadre de la programmation objet enclenche mécaniquement la création de la `vtable` commune mais surtout des 550 milliards de `vptr`, ce qui fait immédiatement passer l'empreinte mémoire de 6 Tio à $(3 \times 4 + 8) \times 8192^3 = 10$ Tio sur une machine 64-bits pour un gain nul en termes de fonctionnalités. L'utilisation inconsidérée du polymorphisme peut donc se traduire par un important overhead au niveau de la mémoire. Le second aspect est celui du temps de calcul du fait des indirections. En effet, les fonctions virtuelles exécutées ne sont plus connues à la compilation mais à l'exécution. Entre l'appel et l'exécution est introduite une couche d'indirection qui est particulièrement destructrice de performances lorsqu'il s'agit d'effectuer des petites opérations des millions ou des milliards de fois comme cela peut être le cas pour des simulations haute performance. Il peut exister un très petit avantage à la virtualité du point de vue du cache lorsqu'il s'agit d'effectuer des branchements à répétition, mais cela est très largement compensé par l'absence d'optimisation. En effet, comme l'appel final ne peut être déterminé à la compilation cela empêche toute possibilité d'inlining [6] et d'optimisation avancée. L'orienté objet peut donc être utile pour tous les aspects de gestion des programmes qui sont exécutés au démarrage ou à la finalisation de l'exécution comme le chargement des fichiers de conditions initiales par exemple, mais il est en général à bannir partout où des boucles sont impliquées et où des optimisations sont nécessaires. Des recherches sont en cours dans les équipes de développement des compilateurs pour procéder à de la dévirtualisation afin de réduire les problèmes de mémoire et de temps de calcul lorsque toutes les informations sont connues à la compilation, mais il s'agit encore de techniques très jeunes et elles sont, de toute façon, destinées à réduire la portée du problème et non pas à le supprimer.

4. Classe principale qui peut être ensuite dérivée en de multiples classes filles selon les principes d'héritages et de polymorphisme.
5. Conversion implicite en une référence vers la classe de base.
6. Le corps des fonctions qui sont « inlinées » est directement insérée dans le code au moment de l'appel ce qui permet d'éviter le surcoût lié à ce dernier.

Le fait d'éviter la programmation orientée objet pour le calcul haute performance est donc complètement motivée par le fait qu'elle empêche la plupart des optimisations à la compilation. L'idéal serait de pouvoir profiter de toutes les améliorations qu'elle propose en termes d'abstraction et en terme d'architecture logicielle, mais sans aucun coût à l'exécution. C'est là qu'interviennent la programmation générique et la programmation générative.

Templates et programmation générique

La généricité à la compilation peut être traitée, en C++ à l'aide des templates[7]. D'autres langages disposent de leur propre système de types génériques, mais dans la suite on se concentre sur des solutions basées sur les choix techniques effectués au début de cette section. Le problème de la fonction `swap` peut être aisément résolu avec des templates :

```
1 template <class T>
2 void swap(T& x, T& y)
3 {
4   const T z = x;
5   x = y;
6   y = z;
7 }
```

Dans ce cas, `T` est remplacé à la compilation par tous les types effectivement utilisés dans le code. À ce stade, il est important de remarquer trois choses. Premièrement, la fonction `swap` n'existe pas vraiment jusqu'à l'étape de compilation : l'extrait de code précédent n'est qu'un patron de fonction et pas une fonction. C'est à la compilation que toutes les versions `swap<int>`, `swap<char>`, `swap<double>` sont générées en fonction de ce qui est effectivement trouvé dans le code : c'est en ce sens que l'on parle de programmation générative. Deuxièmement, si l'utilisateur crée son propre type `my_class`, la version `swap<my_class>` sera générée, toujours suivant le même schéma, à la compilation : c'est en ce sens que l'on parle de programmation générique. Un code source très léger peut ainsi donner naissance à un exécutable de taille importante dans le cas où l'utilisateur crée par exemple des centaines ou milliers de types particuliers puisque chaque type engendre sa propre fonction. Troisièmement, ce patron de fonction `swap` résous le problème de typage posé par la solution à base de pointeurs génériques `void*` : les valeurs sont échangées seulement si elles sont du même type, conformément à ce qui est spécifié par le patron. Ainsi, le code ci-dessous, qui constitue un exemple simple :

```
1 int i = 4;
2 double d = 8;
3 swap(i, d);
```

génèrera une erreur à la compilation puisque le type de `i` et le type de `d` ne sont pas les mêmes. Pour apporter une solution à ce cas, il suffirait d'écrire :

```
1 template <class T1, class T2>
2 void swap(T1& x, T2& y)
3 {
4   const T1 z = x;
5   x = y;
6   y = z;
7 }
```

mais l'on se retrouverait alors encore une fois avec un problème de typage : si échanger deux valeurs arithmétiques a un sens, échanger un entier et une chaîne de caractères en a déjà beaucoup moins. Cette fois, c'est par exemple la version suivante qui entraînerait une erreur :

```
1 int i = 4;
2 std::string s = "eight";
```

7. Types génériques

```
3 swap(i, s);
```

puisque le corps de la fonction n'aurait alors pas de sens. Mais contrairement au cas du `void*`, le problème peut être directement détecté à la compilation, au moment où la spécialisation de l'entier et de la chaîne `swap<int, std::string>` est générée.

Quoi qu'il en soit, l'important pour le calcul haute performance est que les templates peuvent être utilisés pour générer du code, et n'ont, contrairement à la virtualité, aucun impact à l'exécution. En plus des fonctions, il est tout à fait possible de créer des patrons de classe, c'est-à-dire de créer des dépendances de types en fonction d'autres. C'est ce mécanisme qui est utilisé pour générer les tableaux dynamiques `std::vector`. En effet, `std::vector<int>`, `std::vector<char>`, `std::vector<double>` ou `std::vector<my_class>` génèrent tous leur propre version de tableaux dynamiques pour le type spécifié : il ne s'agit pas d'une fonction qui dépend d'un type, mais d'un type, en l'occurrence, `std::vector` qui dépend d'un autre type, en l'occurrence `int`, `char`, `double` ou `my_class`. En plus des types, les templates gèrent les nombres entiers. Les tableaux de taille constante `std::array` en font usage : `std::array<int, 3>` crée ainsi un tableau de trois entiers tandis que `std::array<double, 4>` créé un tableau de quatre flottants en double précision. Il n'est donc plus nécessaire que de spécifier un seul patron, et toutes les versions optimisées pour tous les cas effectivement utilisés sont générées automatiquement à la compilation. Au prix d'un temps de compilation accru et d'un poids d'exécutable possiblement plus important, il est donc possible de garder une grande généricité, de factoriser le code, et de s'assurer des performances maximales, contrairement à ce que permet classiquement l'orienté objet.

Mais ces opérations restent assez basiques puisqu'au final, il ne s'agit que d'utiliser une fonctionnalité du langage pour n'écrire qu'une seule fois le corps d'une fonction ou d'une classe et laisser le compilateur générer toutes les spécialisations. C'est là qu'entre en scène la métaprogrammation qui va permettre d'aller beaucoup plus loin.

5.5.3 Métaprogrammation

Principe

La métaprogrammation fait partie de ces techniques qui n'étaient pas prévus au moment de l'incorporation des templates dans le langage. Elle a été découverte lors du processus de normalisation du C++ [8]. Il s'est en effet avéré que le système de templates était Turing-complet et qu'il était donc possible de l'utiliser pour effectuer des calculs et plus généralement tout ce dont un langage de programmation au sens de Turing est capable d'effectuer. Cette utilisation du système de templates a donné naissance à ce que l'on appelle aujourd'hui le « Template Meta-Programming », ou TMP. Cette dernière regroupe un ensemble de techniques destinées à exécuter du code au sein même du compilateur, souvent en détournant des fonctionnalités du langage de leur usage premier. Dans le principe, il s'agit donc de créer des programmes qui vont être exécutés au moment de la compilation pour effectuer des calculs sur des valeurs ou même sur des types et générer du code, qui lui sera exécuté au lancement du programme. En d'autres termes, des programmes qui génèrent des programmes au sein d'un seul et même langage, avec comme avantage un coût nul à l'exécution.

Illustration par l'exemple : factorielle et fonction d'Ackermann

La meilleure manière d'introduire le concept de métaprogrammation est peut-être encore de présenter un exemple. Le calcul d'une factorielle constitue un exemple standard. De façon classique, la factorielle peut s'implémenter de façon itérative :

```
1 unsigned int iterative_factorial(unsigned int n)
2 {
```

8. L'un des premiers métaprogrammes connus est l'œuvre d'Erwin Unruh en 1994.

```
3   unsigned int fact = 1;
4   while (n > 0) fact *= n--;
5   return fact;
6 }
```

ou de façon récursive :

```
1 unsigned int recursive_factorial(unsigned int n)
2 {
3     if (n > 0) return n*recursive_factorial(n-1);
4     else return 1;
5 }
```

Ces deux implémentations constituent des implémentations standards qui s'exécutent, comme n'importe qu'elle fonction, à l'exécution du programme. Un métaprogramme équivalent à base de structure peut s'écrire :

```
1  template <unsigned int N>
2  struct struct_factorial
3  {
4    static const unsigned int value =
5      N*struct_factorial<N-1>::value;
6  };
7
8  template <>
9  struct struct_factorial<0>
10 {
11   static const unsigned int value = 1;
12 };
```

Dans ce cas, l'appel d'un `struct_factorial<4>::value` à la compilation va nécessiter de la part du compilateur la génération du corps de la structure `struct_factorial<3>` qui va lui-même nécessiter `struct_factorial<2>` puis `struct_factorial<1>` et enfin `struct_factorial<0>` pour lequel la spécialisation a été explicitement fournie en vue de clore le processus récursif. Cela implique donc de la génération de code récursive pour le compilateur, mais une fois la valeur calculée, celle-ci peut être remplacée dans le code généré. Autrement dit, lorsque `N` est connu à la compilation, le coût à l'exécution est nul puisque le résultat est directement calculé par le compilateur. Au lieu d'utiliser des structures, il est possible d'utiliser des fonctions template :

```
1 template <unsigned int N>
2 constexpr unsigned int meta_factorial()
3 {
4     return N > 0 ? N*meta_factorial<N-(N > 0)>() : 1;
5 }
```

où `N-(N > 0)` fournit la condition d'arrêt sans avoir besoin de spécialisation. En effet, la subtilité est que tous les branchements présents dans le corps des fonctions nécessitent d'être générés par le compilateur, même s'ils ne sont pas utilisés. C'est pourquoi, remplacer `N-(N > 0)` par `N-1` entraîne un processus récursif infini : il s'agit donc soit de spécialiser les fonctions template pour stopper la récursion, soit de clore le processus récursif avec des astuces de ce type.

Il est important de noter que la métaprogrammation par template est un langage fonctionnel, tel que peut l'être Haskell : les boucles `for` et `while` doivent être remplacées par une approche purement récursive et les conditions du type `if`, `else if`, `else` doivent être remplacées par de la spécialisation et par le recours à l'opérateur ternaire `?:`. Toutefois, depuis 2011 et l'introduction des expressions constantes `constexpr` et plus encore depuis 2014 avec la relaxation des contraintes appliquées à ces dernières[128], il devrait être possible à moyen terme d'effectuer de plus en plus de calculs au sein du

compilateur sans avoir systématiquement recours à la programmation fonctionnelle.

Puisque c'est bien de cela dont il s'agit ici : transférer du temps de calcul de l'exécution à la compilation. Ce transfert peut être aisément illustré via l'exemple pathologique de la fonction d'Ackermann-Péter[129] qui n'est pas récursive primitive :

$$A(m,n) = \begin{cases} n+1 & \text{si } m = 0 \\ A(m-1,1) & \text{si } m > 0 \text{ et } n = 0 \\ A(m-1, A(m,n-1)) & \text{si } m > 0 \text{ et } n > 0 \end{cases} \tag{5.3}$$

L'implémentation classique de cette fonction pour un calcul au runtime peut s'écrire :

```
unsigned int run_ackermann(unsigned int m, unsigned int n)
{
    if (m == 0) return n+1;
    else if (n == 0) return run_ackermann(m-1, 1);
    else return run_ackermann(m-1, run_ackermann(m, n-1));
}
```

Un métaprogramme équivalent est :

```
template <unsigned int M, unsigned int N>
struct struct_ackermann
{
  static const unsigned int value =
    struct_ackermann<M-1,
      struct_ackermann<M, N-1>::value>::value;
};

template <unsigned int M>
struct struct_ackermann<M, 0>
{
  static const unsigned int value =
    struct_ackermann<M-1, 1>::value;
};

template <unsigned int N>
struct struct_ackermann<0, N>
{
  static const unsigned int value = N+1;
};
```

ou encore, en termes de métafonction constexpr :

```
constexpr unsigned int constexpr_ackermann(unsigned int m,
                                           unsigned int n)
{
    return !m ? n+1 : !n ? constexpr_ackermann(m-1, 1)
      : constexpr_ackermann(m-1, constexpr_ackermann(m, n-1));
}

template <unsigned int M, unsigned int N>
constexpr unsigned int meta_ackermann()
{
    return std::integral_constant<unsigned int,
      constexpr_ackermann(M, N)>::value;
```

Ackermann function computation time		
	Compilation time	**Execution time**
`run_ackermann(4, 1)`	0.34 s	2.99 s
`struct_ackermann<4, 1>::value`	37.85 s	< 0.01 s
`meta_ackermann<4, 1>()`	0.58 s	< 0.01 s

Figure 54 – Temps de calcul de la fonction d'Ackermann. L'exemple de la fonction d'Ackermann permet à la fois d'illustrer l'avantage de la métaprogrammation pour le temps d'exécution et de tester les performances des compilateurs. En l'occurrence GCC 4.8.1, ICPC 14.0.3 et Clang 3.2.1 ont été testés, et seul le premier parvient à calculer $A(4,1)$. Les résultats correspondent aux moyennes des temps utilisateurs donnés par la commande `time` pour 100 compilations et exécutions sur un ordinateur doté d'un processeur Intel I7-2630QM. Les programmes ont été compilés via la commande `time g++ -Wall -Wextra -O3 -std=c++11 -ftemplate-depth=65536 -fconstexpr-depth=65536 ackermann.cpp -o ackermann`. Lorsque N et M sont connus à la compilation, les gains apportés par la métaprogrammation sont très élevés puisque tous les calculs sont effectués en amont par le compilateur. Cela peut considérablement augmenter les temps de compilation comme dans le cas de `struct_ackermann<4, 1>` : dans le cadre du calcul haute performance, cela reste toutefois une bien meilleure solution que l'approche classique puisque les temps de compilation sont généralement négligeables par rapport aux temps d'exécution. L'utilisation des expressions constantes permet de limiter cette augmentation comme l'illustre le temps de compilation de `meta_ackermann<4, 1>()`. Dans des langages comme le Fortran qui ne permettent pas d'intervenir sur le processus de compilation, seule la première approche est disponible.

```
13 }
```

où `meta_ackermann` force l'exécution de `constexpr_ackermann` à la compilation. Cela permet de tester d'une part les avantages de l'approche TMP, mais également de tester les compilateurs eux-mêmes. Et force est de constater que, de ce point de vue, seul GCC s'avère être capable de calculer $A(4,1)$ [9] face aux compilateurs d'Intel et LLVM. Les résultats des temps de compilation et d'exécution sont donnés en figure 54. Il apparait clairement que la métaprogrammation permet de transférer tout le temps de calcul de l'exécution à la compilation.

Métaprogrammation sur les valeurs et sur les types

Ces deux exemples très simples permettent d'illustrer l'approche TMP. Mais cela va bien plus loin que le calcul de constantes à la compilation. En fait, on peut généralement distinguer deux grandes classes de techniques de métaprogrammation : la métaprogrammation sur des valeurs et la métaprogrammation sur des types. Dans la première catégorie se trouvent toutes les techniques qui visent à optimiser des calculs : cela va du calcul de constantes comme dans les deux exemples donnés plus haut, à de l'évaluation paresseuse en passant par de la simplification d'expressions. L'objectif est de précalculer à la compilation tout ce qu'il est possible de précalculer pour ne laisser qu'un nombre d'opérations minimal à l'exécution. Dans ce domaine, l'« expression template » a fait ses preuves et est aujourd'hui massivement utilisé pour optimiser les bibliothèques d'algèbre linéaire. Le principe consiste à générer un arbre syntaxique abstrait (AST) en composant récursivement des types correspondant aux opérations à effectuer, mais ne les effectuant pas. Il est ainsi possible de retarder au maximum l'évaluation et de générer des boucles uniques sans objet temporaire pour des expressions matricielles. La métaprogrammation par template permet aussi par exemple de générer récursivement tous les termes d'un développement en série jusqu'à un ordre fixé à la compilation et de simplifier au maximum l'expression obtenue. Ou d'effectuer à la compilation des masquages sur des bits pour des opérations bit à bit. Ou encore de réarranger et de réaligner des données pour préparer le terrain de la vectorisation. Dans le même esprit, il est possible de faire du dépliage automatique de boucles. Pour résumer, les possibilités d'optimisation sont très grandes et peuvent parfois mener, dans des cas très spécifiques, à des gains de deux ou trois ordres de grandeurs par rapport à des programmes classiques en C ou en Fortran. Il ne s'agira pas, dans la suite, d'exposer toutes ces techniques dans leurs moindres détails, malgré le fait que

9. En enclenchant une profondeur de récursion `-ftemplate-depth=65536 -fconstexpr-depth=65536`

la documentation sur le sujet soit assez éparse[130, 124]. Certaines d'entre elles seront cependant utilisées pour optimiser certains calculs dans le cadre du développement de la bibliothèque MAGRATHEA.

En plus de permettre d'optimiser des calculs, la métaprogrammation permet d'effectuer des opérations sur les types eux-mêmes. Dans ce cas, l'objectif est souvent d'assurer un maximum de généricité avec une factorisation maximale du code. En utilisant ces approches, il est possible d'injecter du code dans du code utilisateur, de concevoir des objets capables de s'adapter à ceux fournis par l'utilisateur, de contrôler le processus d'instanciation, et même, en poussant le concept à son extrême limite, de recréer un compilateur au sein d'un compilateur. Cette dernière possibilité est par exemple mise en œuvre dans BOOST.PROTO qui vise à fournir les outils nécessaires pour d'une part redéfinir des grammaires pour des langages embarqués et d'autre part pour reconstruire et manipuler les AST liés à ces grammaires. Mais sans aller jusqu'à reconstruire un compilateur, ce qui est faisable mais ce qui se situe au-delà de ce qui sera utilisé dans la suite, il est possible de mettre en place du code capable d'assembler et de rétroagir sur des éléments fournis par l'utilisateur par le biais, entre autres, de classes de traits, de classes de politiques et de techniques comme le SFINAE ou le CRTP. À terme, la normalisation d'outils de réflexion à la compilation, que ce soit pour la partie liée à l'introspection ou pour la partie liée à la réification devrait permette de décupler les possibilités de toutes ces approches dans le domaine de la programmation générative.

Deux techniques utilisées : SFINAE et CRTP

Pour comprendre comment il est possible d'intervenir dans le processus d'instanciation des fonctions et classes templates, il est nécessaire de présenter deux techniques essentielles : le SFINAE, pour « Substitution Failure Is Not An Error » et le CRTP, pour « Curiously Recurring Template Pattern ». La première technique est particulièrement utile pour effectuer de l'introspection au moment de la compilation et inférer les propriétés de certains types. L'élément central est qu'un paramètre template qui est invalide du point de vue de la compilation, n'est pas une erreur du point de vue de la syntaxe du langage. En créant volontairement des problèmes de substitution on peut ainsi maîtriser l'instanciation via les propriétés des types. Le meilleur moyen d'illustrer le procédé est encore d'utiliser un exemple. Soit une fonction `f` destinée à effectuer la somme de deux variables :

```
1 template <class T>
2 T f(const T& x, const T& y)
3 {
4   return x+y;
5 }
```

Cette fonction sera instanciée pour tous les types `T` et si l'opérateur + n'a pas été défini pour `T`, le compilateur génèrera un message d'erreur car dans ce cas, le corps de la fonction n'a pas de sens. C'est par exemple le cas pour un tableau de 3 réels : `T = std::array<double, 3>`. Admettons que l'on souhaite maintenant généraliser l'approche aux tableaux de 3 réels. Il suffit d'ajouter la surcharge adéquate :

```
1 std::array<double, 3> f(const std::array<double, 3>& x,
2                          const std::array<double, 3>& y)
3 {
4   return {{x[0]+y[0], x[1]+y[1], x[2]+y[2]}};
5 }
```

Donc si le type est un tableau de 3 réels, le compilateur commencera par tester si la surcharge spécialisée convient et sinon, il utilisera la version générique. Maintenant, imaginons que l'on souhaite restreindre la version générique aux types arithmétiques et que l'on souhaite généraliser la version spécialisée à tout ce qui pourrait ressembler à un tableau de 3 réels, y compris des types créés par l'utilisateur. La programmation générique standard ne convient alors plus. C'est là que SFINAE entre en jeu. Il va d'abord s'agir de bloquer l'instanciation de la version générique pour tous les types non arithmétiques.

Pour cela on utilise `std::enable_if` qui définit ou non `::type` suivant que l'argument template fourni est vrai ou faux. Dans la pratique cela donne :

```
template <class T, class = typename std::enable_if<
  std::is_arithmetic<T>::value>::type>
T f(const T& x, const T& y)
{
  return x+y;
}
```

Deux possibilités : soit `std::is_arithmetic<T>::value` est `true` et dans ce cas `::type` existe, soit `std::is_arithmetic<T>::value` est `false` et dans ce cas `::type` n'existe pas. Dans le premier cas, le compilateur essaye d'implémenter la fonction, dans le second, cela crée un problème mais qui n'est pas considéré comme une erreur de compilation puisque cela intervient au niveau template et non au niveau du corps de la fonction. Dans ce second cas, le compilateur cherche simplement une surcharge plus spécialisée que la version générique. S'il n'en trouve pas, alors là, effectivement, une erreur de compilation est générée puisque l'appel de la fonction n'a pas de sens. Ainsi, il est possible de bloquer manuellement certaines instanciations pour orienter le compilateur vers les surcharges souhaitées. Pour généraliser la spécialisation à tous les tableaux de 3 réels, on peut procéder de la façon suivante :

```
template <class T, class = typename std::enable_if<
  std::decltype<T>().size() == 3>::type, class = typename
  std::enable_if<std::is_floating_point<typename
  std::decay<declval(std::decltype<T>()[0])>::type>
  ::value>::type>
T f(const T& x, const T& y)
{
  T z;
  for (std::size_t i = 0; i < z.size(); ++i) z[i] = x[i]+y[i];
  return z;
}
```

, la partie template pouvant se traduire par « si T implémente une fonction `size()` constante retournant 3 et si T dispose d'un opérateur `[]` qui retourne un flottant, alors dans ce cas seulement ce qui suit peut être traité comme un cas valide ». Au final, en utilisant la technique SFINAE nous avons restreint la validité du cas générique et généralisé celle du cas particulier. Toutefois cela fonctionne bien parce que l'intersection des contraintes sur T est nulle. Dans le cas où T est à la fois un type arithmétique et un tableau de trois réels, il existe une ambiguïté qui se traduit par une erreur de compilation. L'implémentation de MAGRATHEA a permis de mettre au jour une astuce pour lever l'indétermination via la priorité accordée aux variadic templates. En effet, en remplaçant la première fonction par :

```
template <class T, class... Dummy, class =
  typename std::enable_if<std::is_arithmetic<T>::value
  && !sizeof...(Dummy)>::type>
T f(const T& x, const T& y, Dummy...)
{
  return x+y;
}
```

on ordonne la priorité des surcharges : en l'occurrence, elle passe au second plan. Le compilateur tentera donc d'abord d'instancier l'autre version avant de revenir sur celle-ci. En procédant de la sorte et en généralisant cette mécanique à des types plus complexes, il est donc possible de contrôler exactement le « chemin »suivi par le compilateur pour générer le code de l'exécutable. Cela permet un contrôle très fin de la généricité en fonction des propriétés des types, y compris des nouveaux types créés par l'utilisateur.

À cela s'ajoute le « Curiously Recurring Template Pattern » et l'astuce de Barton–Nackman[131] qui permettent aisément d'injecter du code dans des classes et qui offrent les mêmes possibilités que celles offertes par le polymorphisme dynamique de l'orienté objet, mais de façon statique, à la compilation. Le principe consiste à passer les classes dérivées aux classes dont elles héritent :

```
template <class CRTP>
struct Base
{
  void base_function()
  {
    static_cast<CRTP&>(*this).derived_function();
  }
  static void static_base_function()
  {
    CRTP::static_derived_function();
  }
};

struct Derived : public Base<Derived>
{
  void derived_function();
  static void static_derived_function();
};
```

Ce faisant, il est possible pour la classe de base, d'accéder à tous les membres des classes dérivées. En couplant cette approche avec des classes de traits, on peut ainsi créer des véritables générateurs de code capables d'étendre les fonctionnalités des types utilisateurs. On peut par exemple utiliser le CRTP pour ajouter automatiquement des fonctionnalités d'algèbre linéaire à des tableaux créés par l'utilisateur, comme on pourrait le faire par de l'héritage classique, mais sans indirection, tout étant pris en charge par le compilateur. Avec de l'héritage multiple, on met ainsi à disposition de l'utilisateur une série d'outils capables d'apporter automatiquement les propriétés désirées à des objets nouvellement créés.

Évolutions futures : concepts

En 2017, le nouveau standard du langage devrait apporter le support des concepts[126], dont la normalisation avait échouée en 2011. GCC en offre déjà un support partiel. Il s'agit d'asseoir SFINAE sur des bases plus solides en évitant une syntaxe verbeuse et des messages de compilation parfois obscurs. L'objectif est de pouvoir contraindre les templates et permettre l'écriture de :

```
template <Arithmetic T>
T f(const T& x, const T& y)
{
  return x+y;
}
```

lorsque le type `T` doit être un type arithmétique et où `Arithmetic` désigne un concept. Contrairement à SFINAE, cela permettra une gestion native des intersections non nulles de contraintes pour déterminer la priorité d'instanciation. Dans l'absolu, tout ce qu'il est possible de faire avec les concepts sous la forme attendue pour 2017, peut être fait actuellement avec SFINAE, de la spécialisation et l'utilisation des variadic templates. Tout code reposant massivement sur des `std::enable_if` devrait ainsi pouvoir être facilement mis à niveau pour tirer parti de la nouvelle syntaxe.

5.6 Application à la cosmologie

5.6.1 Vers des bibliothèques actives pour la cosmologie : l'exemple de MAGRATHEA

Introduction

Comme nous l'avons vu dans les sections précédentes, des problèmes fondamentaux de conception sont posés par les architectures logicielles actuelles du point de vue de la complexité des codes que seuls des compilateurs ou des langages dédiés semblent à même de résoudre. Or, la métaprogrammation permet justement d'adapter le comportement des compilateurs pour effectuer des opérations, que ce soit sur des valeurs ou sur des types, pour générer ou réarranger du code. Ce faisant, il est possible d'atteindre un double objectif : celui de considérablement simplifier les codes par une approche générique en laissant les compilateurs traiter l'explosion combinatoire et celui d'atteindre des performances maximales dans un contexte de calcul haute performance. L'un des inconvénients majeurs est un temps de conception accru du fait du caractère parfois peu intuitif des méthodes décrites dans la section précédente ainsi que de la difficulté de débogage, car les erreurs liées à la métaprogrammation se produisent à la compilation et non pas à l'exécution. Pour tenter d'adapter ces approches, pour l'instant très théoriques, au contexte de la cosmologie numérique, le développement d'une bibliothèque expérimentale MAGRATHEA [10] a été initié. Il s'agissait également de fournir des outils d'analyse pour le Full Universe Run, notamment en ce qui concerne la propagation de la lumière. Les paragraphes suivants illustrent des aspects clefs de cette implémentation aujourd'hui pleinement fonctionnelle.

Algèbre linéaire

La métaprogrammation peut d'abord être utilisée pour optimiser des opérations d'algèbre linéaire, utile par exemple pour représenter des vecteurs et les opérations sur ces vecteurs. Il ne s'agissait pas ici de réexpérimenter toutes les techniques d'expression template utilisées dans des bibliothèques comme EIGEN[123], mais plutôt voir comment il était possible de mettre en œuvre de la génération de code pour que l'utilisateur puisse spécifier ses propres types de données et les étendre à volonté. Dans le contexte de la cosmologie cela peut être utile pour générer des types de bases comme des quantités scalaires, des points, des vecteurs, ou des tenseurs. Un problème souvent rencontré avec les bibliothèques d'algèbre linéaire classiques est qu'elles viennent avec leurs propres opérations et qu'il peut être difficile d'en tirer parti pour des nouveaux types de données. En effet, l'héritage peut poser problème du point de vue de la destruction des objets si les types fournis n'ont pas été prévu pour cela à l'origine, et la composition peut donner lieu à une très grande redondance du code s'il s'agit d'exposer toutes les méthodes des types composés. C'est là où l'usage du CRTP peut être très pertinent : au lieu de fournir des types de base, MAGRATHEA fournit les outils pour générer ces types de base. Trois outils sont au cœur de ce mécanisme :

```
1  class vectorizer
2  {/*...*/};
3
4  template <class Kind,
5            std::size_t Size,
6            template <class, Kind...> class Crtp,
7            class Type,
8            Kind... Parameters>
9  class static_vectorizer : public vectorizer
10 {/*...*/};
11
12 template <class Kind,
13           template <class, Kind...> class Crtp,
```

10. Multi-processor Adaptive Grid Refinement Analysis for THEoretical Astrophysics

```
14              class Type,
15              Kind... Parameters>
16 class dynamic_vectorizer : public vectorizer
17 {/*...*/};
```

un `vectorizer` pour disposer d'une racine commune lorsqu'il s'agit d'appliquer des contraintes à l'aide de `std::is_base_of`, un `static_vectorizer` pour générer automatiquement toutes les opérations vectorielles sur un type de taille constante créé par l'utilisateur et un `dynamic_vectorizer` pour en faire de même sur les types de taille variable. Aussi, pour générer un nouveau type à partir de ces outils, il suffit d'écrire par exemple :

```
1 template <class T, std::size_t N>
2 class my_vector
3 : public static_vectorizer<std::size_t, N, my_vector, T, N>
4 {/*...*/}
```

et de fournir un unique opérateur `[]`. Les milliers de lignes nécessaires pour gérer toutes les opérations possibles avec des quantités scalaires et vectorielles de même type ou de type hétérogène sont alors générées automatiquement pour `my_vector` via le CRTP. Il est ainsi possible de fournir un ensemble de générateurs que l'utilisateur peut composer pour offrir toutes les fonctionnalités qu'il souhaite à ses objets de base tout en lui laissant la possibilité d'ajouter ses propres fonctionnalités. On évite ainsi intelligemment l'antipattern du « god object » [11] pour des éléments devant pourtant implémenter un grand nombre de caractéristiques et d'algorithmes. Ces outils illustrent parfaitement le fait qu'une bibliothèque active peut accomplir ce dont une bibliothèque passive classique est incapable : rétroagir sur le code utilisateur au moment de la compilation.

Maillage

Un élément central des codes de simulation est le maillage. Dans toute la suite on s'intéresse au cas d'un maillage adaptatif cubique périodique, comme dans RAMSES. L'un des problèmes de RAMSES justement est le manque de généricité du maillage. Ajouter ou supprimer des variables physiques au sein de ce dernier peut impliquer la modification de plus d'une dizaine de fichiers et de multiples duplications de code. L'idéal serait de disposer d'un maillage totalement indépendant de son contenu et des opérations qui sont effectuées sur ce contenu. Une telle piste a été explorée au cours de l'implémentation de MAGRATHEA. Pour illustrer ce qu'il est possible de faire par l'intermédiaire de la programmation générative, étudions le cas d'une opération relativement simple : un « Cloud-In-Cell » inverse, noté dans la suite CIC^{-1}, pour interpoler des quantités distribuées sur une grille à l'emplacement de particules. Une telle interpolation dépend de plusieurs choses :

- la dimensionnalité du problème
- le type des particules
- le type des données distribuées dans la grille

La métaprogrammation permet d'implémenter un maillage capable de s'adapter automatiquement à ces trois caractéristiques via un template de la forme :

```
1 template <std::size_t Dimension,
2           class Field,
3           class Particle,
4           /*...*/>
5 class mesh
6 {/*...*/};
```

Tout d'abord, au lieu de se limiter à 3 dimensions, ce qui est le cas de la plupart des codes de cosmologie, on peut travailler en dimension arbitraire n, cette dernière étant spécifiée au moment de la

11. Tendance à créer des objets disposant de trop de fonctionnalités pour traiter tous les cas possibles et imaginables.

compilation. À partir de cette dernière, on peut calculer récursivement en TMP le nombre d'hypercubes m-dimensionnels à la frontière d'un hypercube n-dimensionnel :

$$E_{m,n} = 2^{n-m} \binom{n}{m} = 2^{n-m} \frac{n!}{m!\,(n-m)!} \tag{5.4}$$

ce qui donne par exemple le nombre de faces d'un cube tridimensionnel $E_{2,3} = 6$. Le résultat nous renseigne exactement sur la manière dont le CIC inverse doit procéder pour itérer sur les hypercubes voisins au sein de la grille en vue de récupérer les valeurs à interpoler. La forme de la procédure itérative est donc entièrement connue à la compilation, ce qui permet, en plus de fournir une grande généricité, d'atteindre de très bonnes performances puisque le compilateur peut procéder à des optimisations telles que de la vectorisation. Une fois le processus itératif connu, le même type d'approche peut être mis en œuvre pour l'interpolation, à la condition que les données contenues dans la grille `Field` disposent d'une interface commune, à la manière des `std::tuple`. Ainsi, si chaque élément de grille contient par exemple une chaîne de caractères, une quantité scalaire, une quantité vectorielle, et un entier :

```
1 std::tuple<std::string, double, std::array<double, 3>, int>
```

il suffit d'extraire les éléments les uns après les autres récursivement à l'aide d'une fonction variadic template [12] et d'en calculer la moyenne s'il s'agit de quantités arithmétiques ou vectorielles. Le tri entre ce qui peut faire l'objet d'une interpolation et ce qui ne peut pas en faire l'objet peut être réalisé via du SFINAE. Lorsque la réflexion statique sera en place dans le langage, il ne sera même plus nécessaire que les quantités disposent d'une interface commune. À ce stade la moyenne pondérée du contenu arbitraire de la grille a été effectuée sur un nombre arbitraire de dimensions : il ne reste plus qu'à attribuer la valeur obtenue à la particule. Là, encore, il suffit que les particules disposent d'une interface commune pour leur affecter le résultat : cela peut être par exemple l'opérateur `[i]` pour `i` allant de `0` à `Dimension-1`. L'important d'un point de vue de la généricité est qu'il n'est pas nécessaire de connaître le type pour appeler l'opérateur `[]` : l'utilisateur final peut donc spécifier le type de particules qu'il souhaite tant qu'il fournit les opérateurs nécessaires. La validité de l'interface, peut être, encore une fois, contrôlée au moyen de la technique SFINAE. Cette façon de procéder conduit ainsi à un algorithme de CIC inverse capable de travailler sur n'importe quel maillage, quel que soit sa dimension ou son contenu.

Particules, champs et quantités intégrées

Le maillage, on l'a vu, nécessite deux types de grandeurs : des grandeurs de type particule et des grandeurs de type champ. Les premières évoluent au sein du maillage tandis que les seconds restent fixes par rapport à celui-ci. Certains codes de cosmologie ne font appel qu'aux premières tandis que d'autres ne font appel qu'aux seconds, dans le cas de simulations purement hydrodynamique par exemple. Certains, enfin, font appel conjointement aux deux éléments. Concevoir des abstractions pour ces quantités est, dans ce cadre, essentiel, car elles occupent une place tout à fait centrale dans les simulations cosmologiques. Dans la pratique, il peut s'agir par exemple de spécifier le contenu physique d'une cellule hydrodynamique en termes de pression P, de densité ρ ou de température T, ou de spécifier un type de particules caractérisé par son vecteur position $\vec{x}$, sa vitesse $\vec{v}$, sa masse m et la force qu'elle subit $\vec{f}$. Dans une approche classique, l'ajout d'une quantité physique consisterait à modifier tous les champs et les tableaux utilisés pour stocker les données associées dans le code. Dans une approche générique, c'est au compilateur qu'il incombe la tâche d'opérer les modifications adéquates dans toutes les parties du code concernées.

Une première façon de solutionner le problème, sous l'angle de la programmation orientée objet, serait de concevoir des classes abstraitres pour les particules et les champs, qu'il conviendrait ensuite de dériver. Seulement voilà, comme cela a déjà été précisé, la virtualité entraîne un surcoût mémoire

12. Fonction disposant d'un nombre arbitraire d'arguments.

conséquent. Sans compter les indirections éventuelles pour chaque fonction appelée sur ces données qui sont au cœur du calcul. Il s'agit là d'un coût prohibitif dans un contexte de calcul haute performance. Une autre approche reposerait sur la simple utilisation de structures de données à membres publics. Mais sans réflexivité du contenu des structures il est impossible de mettre en place une démarche générique. Reste alors la voie des n-uplets, ou plus généralement celle des classes apparentées aux `tuple` : `std::tuple`, `std::array` et `std::pair`, même s'il est pour l'instant difficile de conceptualiser cet ensemble d'objets au sein du langage. En utilisant cette approche, il devient possible d'extraire récursivement les types fondamentaux des types composites via des templates variadiques et d'effectuer les opérations désirées sur ces types fondamentaux via SFINAE. Dans le cas concret d'un `std::tuple<std::string, double, double>` spécifiant une chaîne de caractère et deux réels pour chaque cellule par exemple, et pour une opération d'interpolation de Cloud-In-Cell inverse, il est ainsi possible d'opérer récursivement sur chacune des variables et de n'effectuer l'interpolation que dans le cas où celles-ci sont réelles. De cette façon, il est possible de concevoir l'ensemble des algorithmes de façon générique et laisser au compilateur le soin de générer les bonnes fonctions optimisées adaptées à chaque cas, c'est-à-dire à chaque type de particules, de champs et de quantités intégrées.

Mais ne faire reposer la généricité que sur des `std::tuple`, `std::array` et des `std::pair` est assez limité. C'est pourquoi, le concept de « tuple-like class » a été forgé. Une classe « tuple-like » au sens de Magrathea est soit une classe dont le nombre de composantes internes est donné par `std::tuple_size`, lesquelles sont accessibles via `std::get` soit une classe qui dispose d'une fonction membre `data` renvoyant une référence vers une donnée membre dont le nombre de composantes internes est donné par `std::tuple_size`, lesquelles sont accessibles via `std::get`. Il s'agit en quelque sorte d'une façon de généraliser les n-uplets de façon à les enrichir de davantage de fonctionnalités. Cela est par exemple utile pour proposer une abstraction des quantités intégrées. En effet, en physique, la résolution d'équations différentielles ordinaires, ou d'équations aux dérivées partielles est assez centrale, et les algorithmes d'intégrations sont indépendants des quantités intégrées. On peut ainsi concevoir un intégrateur de Runge–Kutta–Fehlberg d'ordre 4/5 générique et prenant en entrée le vecteur à intégrer, les équations à résoudre, les conditions d'arrêt et les critères de sauvegarde des résultats. Au final, un tel intégrateur génère une table avec la valeur du vecteur intégré à chaque étape ainsi qu'éventuellement des quantités dérivées du vecteur intégré. Pour être le plus générique possible, Magrathea propose donc le concept de quantité intégrée. Une quantité intégrée est une classe composite de trois classes « tuple-like » désignées par :

- la partie `id` qui identifie de façon unique l'étape d'intégration : il peut simplement s'agir d'un indice entier
- la partie `core` qui est un « tuple-like » itérable et qui représente le vecteur d'intégration lui-même
- la partie `extra` dans laquelle sont stockées d'éventuelles quantités spécifiques à chaque étape d'intégration, mais non primaires dans le sens où elles peuvent être dérivées directement des quantités intégrées

Ces trois types de quantités sont un pilier essentiel pour le reste de la bibliothèque. Ils nous amènent également à réfléchir sur l'importance à disposer à terme d'un concept semblable de « tuple-like » en standard dans le langage, ou mieux, de disposer d'outils de réflexion.

Intégration

Le fait de disposer de quantités génériques pour représenter les résultats d'intégration permet de concevoir, comme énoncé précédemment, des algorithmes d'intégration génériques. Le concept n'est pas nouveau et a conduit à l'élaboration de la bibliothèque Boost.Odeint. On utilise ici une approche du même genre, moins exhaustive, mais répondant pleinement aux besoins en solveur. Pour cela, le vecteur d'intégration, et la dérivée du vecteur d'intégration sont fournis en paramètres templates de même que la condition d'arrêt. Dans le cas d'intégrateur de Runge-Kutta, il est même possible d'aller plus loin en termes de généricité en autorisant la spécification du tableau de Butcher à la compilation.

Entrées et sorties

Lorsqu'il s'agit d'utiliser le C++ dans le contexte du calcul haute performance, se pose l'éternel problème de la bibliothèque standard d'entrée/sortie `iostream`. En effet, contrairement au reste de `std` qui fait appel à la programmation générique, cette partie remontant à des époques reculées de la conception du langage a été largement basée sur une approche orientée objet. Et l'impact en termes de performances s'en ressent. S'il est important d'utiliser des buffers pour écrire et lire les données par paquet en vue d'atteindre des débits optimaux, le problème majeur de `iostream` est que la lecture et l'écriture dans ces buffers passe par un empilement de couches de virtualité. Et dans un contexte haute performance, ces indirections ont un coût significatif. Sur le long terme, le problème sera définitivement réglé lorsque l'introspection et la réification auront été ajoutés au langage : il sera alors possible de concevoir une nouvelle bibliothèque standard de sérialisation avec comme objectif d'atteindre les meilleures performances possibles en termes de débit d'entrées et sorties. En attendant, des outils de base ont été conçus pour Magrathea. Pour écrire ou lire des zones entières de mémoire, que ce soit dans des buffers ou directement sur disque, mais également pour sérialiser des données. En effet, des structures ou des n-uplets ne peuvent être écrits directement en assurant la compatibilité entre systèmes du fait de l'alignement et du padding intra et inter-structures. En conséquence, on utilise dans notre cas une sérialisation reposant sur l'extraction et l'écriture des données par accès récursif aux types fondamentaux depuis des objets composites satisfaisant le concept de « tuple-like » présenté précédemment. Une couche de conversion d'endianness générique est ajoutée pour l'écriture et la lecture :

```
template <class T, class = typename
  std::enable_if<std::is_fundamental<T>::value>::type>
inline void byteswap(T& x)
{
  constexpr int half = sizeof(x)/2;
  constexpr int end = sizeof(x)-1;
  char* buffer = reinterpret_cast<char*>(&x);
  char tmp;
  for (int i = 0; i < half; ++i) {
    tmp = buffer[i];
    buffer[i] = buffer[end-i];
    buffer[end-i] = tmp;
  }
}
```

dont l'activation au sein de toutes les routines dépend d'un paramètre template, ce qui permet au compilateur de nettoyer l'appel lorsqu'aucune conversion n'est nécessaire.

En utilisant ces fonctions de base pour la gestion des entrées/sorties, il est possible de concevoir des classes de lecture optimisées pour de nombreux types de fichier. Dans notre cas, le fait de pouvoir lire et écrire des fichiers binaires, des fichiers au format binaire Fortran, des fichiers textes de type `csv` ou encore des fichiers de paramètres est essentiel et a donc été implémenté au sein de la bibliothèque. Du fait de l'expérience d'archivage du Full Universe Run, des routines de concaténation et de découpage des fichiers ont également été implémentées en vue de proposer des solutions efficaces et performantes de gestion de fichiers adaptées aux limitations des systèmes de stockage des supercalculateurs.

Parallélisation

Un dernier point d'intérêt concernant Magrathea concerne le parallélisme. Pour maximiser la modularité des codes, les algorithmes liés à la parallélisation ont été extraits autant que faire se peut. Le parallélisme en question est hybride et repose sur une approche conjointe MPI/`std::thread`. Á terme, la partie MPI pourrait éventuellement être remplacée par des appels à la future version stan-

dardisée de la bibliothèque réseau Boost.Asio en vue d'autoriser un contrôle fin des communications. Concernant la parallélisation inter-nœud, l'option `std::thread` a été choisie face à OpenMP pour la plus grande liberté de parallélisme de tâche qu'elle permet : contrairement à une approche OpenMP classique qui consiste à répartir l'exécution de boucles sur différents processus, `std::thread` repose sur le détachement de threads de l'exécution principale pour exécuter certaines fonctions particulières. Avec cette approche, il est très facile d'exécuter des fonctions de nature très différente simultanément, ce qui est par exemple utile lorsqu'il s'agit de continuer à effectuer des calculs pendant qu'un thread entre dans une phase intense d'écriture. La parallélisation de boucles est moins intuitive et passe par des algorithmes de ce type :

```
template <class Iterator, class Function>
double parallel_for_each(const Iterator& first,
                         const Iterator& last,
                         Function&& f,
                         const int nthreads = 1)
{
  // Typedefs
  using clock = std::chrono::high_resolution_clock;
  using integer = long long int;
  using duration = std::chrono::duration<double>;

  // Variables
  const clock::time_point tbegin = clock::now();
  const integer ntasks = std::max(static_cast<int>(1),
                                  nthreads);
  const long long int group
    = std::max(static_cast<integer>(first < last),
               static_cast<integer>((last-first)/ntasks));
  std::vector<std::thread> threads;

  // Parallelization
  Iterator it = first;
  threads.reserve(ntasks);
  for (it = first; it < last-group; it += group) {
    threads.push_back(std::thread([=, &last, &group, &f](){
      std::for_each(it, std::min(it+group, last), f);}));
  }
  std::for_each(it, last, f);

  // Finalization
  std::for_each(threads.begin(),
                threads.end(),
                [](std::thread& current){current.join();});
  return std::chrono::duration_cast<duration>(
    clock::now()-tbegin).count();
}
```

où `f` est la fonction à exécuter sur chaque élément de l'ensemble d'itérateurs `[first, last[`. De la même façon, en utilisant des pointeurs sur fonction, des foncteurs et des lambda, il est possible de désolidariser la parallélisation du reste du code.

D'autres illustrations d'utilisation de programmation générique et de programmation générative dans le contexte de la cosmologie numérique pourraient être présentés : on s'arrêtera toutefois là, le

lecteur pouvant se reporter à la documentation de MAGRATHEA lorsque la bibliothèque sera finalisée et publiée sous licence libre pour davantage de détails.

5.6.2 Proposition d'architecture logicielle unifiée pour la cosmologie numérique

Introduction

La sous-section précédente a montré comment les approches de langages embarqués pouvaient être mises en œuvre pour créer des bibliothèques actives pour la cosmologie. Disposant de ce bagage technique et de cette démonstration de faisabilité que sont les travaux exploratoires sur la bibliothèque MAGRATHEA, il est maintenant possible de prendre du recul et de poser les bases d'une architecture logicielle unifiée pour la cosmologie. De la même manière que la physique consiste à mettre en lumière des symétries, la conception de telles architectures repose sur la recherche de symétries et de régularités au sein des codes de cosmologie numérique. La proposition qui suit est ainsi basée sur les leçons du Full Universe Run, la comparaison de plusieurs codes de simulation, les analyses de ces simulations les plus couramment menées, mais également les analyses des grands relevés. Car finalement un certain nombre d'analyses observationnelles aux grandes échelles reposent sur des démarches semblables à celles utilisées pour les simulations.

Distribution spatiale de données : quantités eulériennes et quantités lagrangiennes

Que l'on s'intéresse à des simulations ou à des observations, la première régularité concerne la façon de représenter les données. Et dans la majorité des cas, le principe est toujours le même : ces données sont représentées dans un espace à N dimensions (souvent avec $N = 3$) sous un point de vue eulérien ou sous un point de vue lagrangien. Dans la première catégorie, celle des quantités « eulériennes », on peut ranger toutes les quantités liées à des champs scalaires, vectoriels ou tensoriels échantillonnés dans l'espace au moyen d'un maillage, adaptatif ou non. Il peut par exemple s'agir d'un champ de vitesse échantillonné sur une grille, ou de la valeur de la métrique sur un maillage régulier. Toutes les données de type « champ » présentées dans la sous-section précédente en font partie. En supplément de ce point de vue eulérien, existent des quantités « lagrangiennes ». Il s'agit basiquement de toutes les quantités physiques pouvant être liées à des particules. Dans cette dénomination, une particule de matière noire d'une simulation N-corps est une donnée « lagrangienne ». De même qu'une galaxie au sein d'un grand relevé observationnel.

Il apparait donc essentiel qu'une architecture logicielle pour la cosmologie numérique dispose de ces deux concepts de données eulériennes et de données lagrangiennes. Il s'agit véritablement du cœur de l'architecture présentée dans le paragraphe suivant.

Composantes statiques, cinématiques et dynamiques

Comment faire émerger des régularités au sein des codes de simulation et d'analyse ? Il n'y a pas unicité de la solution, mais une approche intéressante consiste à travailler en niveaux de complexité, de la même manière de ce qui a pu être fait pour BLAS [13], le célèbre ensemble de fonctions standardisées pour l'algèbre linéaire. À quoi ressemblerait donc l'équivalent de BLAS pour la cosmologie numérique ?

En fait, il apparait que tout code de simulation cosmologique peut être divisé en trois niveaux de complexité : une composante statique, une composante cinématique et une composante dynamique.

La composante statique est celle qui travaille sur des quantités eulériennes et des quantités lagrangiennes fixées. Par exemple, calculer des moyennes de quantités dans des zones spatiales fait partie du domaine statique. Ou détecter des structures cosmologiques : dans ce cas il s'agit généralement de

13. Basic Linear Algebra Subprograms

calculer des surdensités locales et de regrouper des données lagrangiennes. Ou encore détecter des vides cosmiques. Ou, dans un domaine numérique ou observationnel, calculer des fonctions de corrélations. Tout cela est basé sur un ensemble de données fixées une fois pour toutes dans l'espace. En comparant de nombreux codes de cosmologie, il apparait également que des structures de données absolument centrales pour la partie statique sont les arbres. Les octrees des maillages adaptatifs ou les kd-trees utilisés pour la détection des voisins sont quelques exemples parmi d'autres. Aussi, du fait de la centralité de cette structure de données, optimiser les codes cosmologiques ne peut faire l'économie d'un travail de fond sur l'optimisation et la généricité des arbres. Gagner quelques pourcents de temps d'exécution sur la navigation dans des arbres ou sur l'occupation mémoire de ces derniers peut ainsi avoir un impact pour l'ensemble de la cosmologie numérique.

La seconde composante est la composante cinématique : dans ce cas on considère des quantités eulériennes fixées, mais des quantités lagrangiennes variables. Autrement dit, il s'agit de faire naviguer des particules dans des champs n'évoluant pas. L'élément essentiel de cette composante, c'est l'intégration numérique d'équations différentielles. Intégrer les équations géodésiques pour un champ gravitationnel arbitraire en approximation de champ faible rentre dans ce cadre cinématique. Comme pour la partie statique, la parallélisation de cette composante n'est généralement pas à l'origine de problématiques très complexes, la distribution spatiale des champs étant connue une fois pour toute.

À cela se rajoute enfin le troisième niveau de complexité : la composante dynamique. Dans ce cadre, les quantités eulériennes et les quantités lagrangiennes sont variables. Tout code complet de formation des structures nécessite ce troisième niveau de complexité. C'est à ce niveau que sont implémentées les méthodes de différences, d'éléments ou de volumes finis. À ce stade, la parallélisation s'avère aussi beaucoup plus complexe puisque le maillage et la charge par processeur sont susceptibles d'évoluer au cours d'une simulation. C'est dans ce contexte qu'il est généralement nécessaire de faire appel aux systèmes de Lindenmayer et aux courbes de remplissage d'espace [14]. C'est aussi dans ce contexte qu'il devient nécessaire d'introduire des zones de recouvrement et des cellules « fantômes » pour effectuer les calculs dynamiques de façon parallèle.

À ce découpage en trois niveaux s'ajoute finalement tout ce qui rentre dans le domaine des utilitaires de calcul :

- la gestion des fichiers
- la parallélisation
- la gestion du temps
- la sauvegarde des données
- le chargement de données et le redémarrage

La représentation graphique de l'ensemble est donné en figure 55.

Mise en pratique et perspectives

Une fois cette architecture commune à la grande majorité des codes cosmologiques formalisée en deux types de données, eulériennes et lagrangiennes, et en trois niveaux de complexité, statique, cinématique et dynamique, la conception de codes génériques, performants, modulaires et extensibles devient d'une part beaucoup plus aisée et d'autre part beaucoup plus lisible pour l'utilisateur final. Il devient ainsi plus facile de créer une sous-partie de code pouvant être étendue par la suite.

Un exemple illustratif de ce que pourrait être une partie statique inspirée de cette architecture est donné en figure 56. La création d'un code dynamique complet sort du cadre de cette thèse, toutefois il

14. Ces aspects seront abordés dans le prochain chapitre.

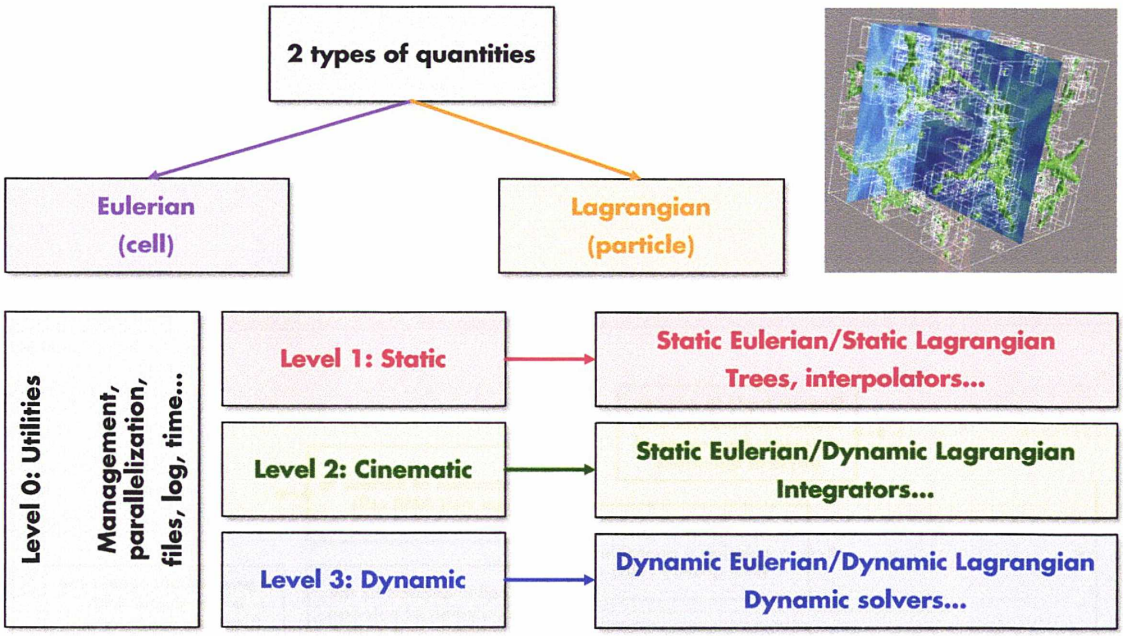


Figure 55 – Architecture logicielle en trois niveaux de complexité pour les codes cosmologiques. L'observation de régularités au sein des codes cosmologiques permet de mettre en lumière la possibilité d'une architecture logicielle centrée sur deux types de quantités, eulériennes et lagrangiennes et trois niveaux de complexité, statique, cinématique et dynamique en plus d'un niveau d'utilitaires. La conception de codes s'inscrivant dans cette optique très générale permet d'une part d'assurer un maximum de modularité et d'autre part un maximum de lisibilité pour l'utilisateur final. Il est aussi plus facile de se concentrer sur certaines composantes des codes tout en gardant en tête la façon dont ces composantes s'inscrivent au sein d'une architecture plus globale.

devient une perspective naturelle des deux chapitres suivants qui se concentrent sur des aspects statiques et cinématiques essentiels pour la résolution des équations géodésiques en champ faible. Aussi, l'utilisation des approches de programmation générative, de programmation générique, et de langages embarqués, couplée à des architectures logicielles bien pensées apparait comme primordiale d'une part pour venir à bout de la complexité actuelles des codes et de leur manque de modularité pourtant nécessaire pour explorer de multiples pistes théoriques, et d'autre part pour amener la cosmologie numérique à l'échelle de l'exascale.

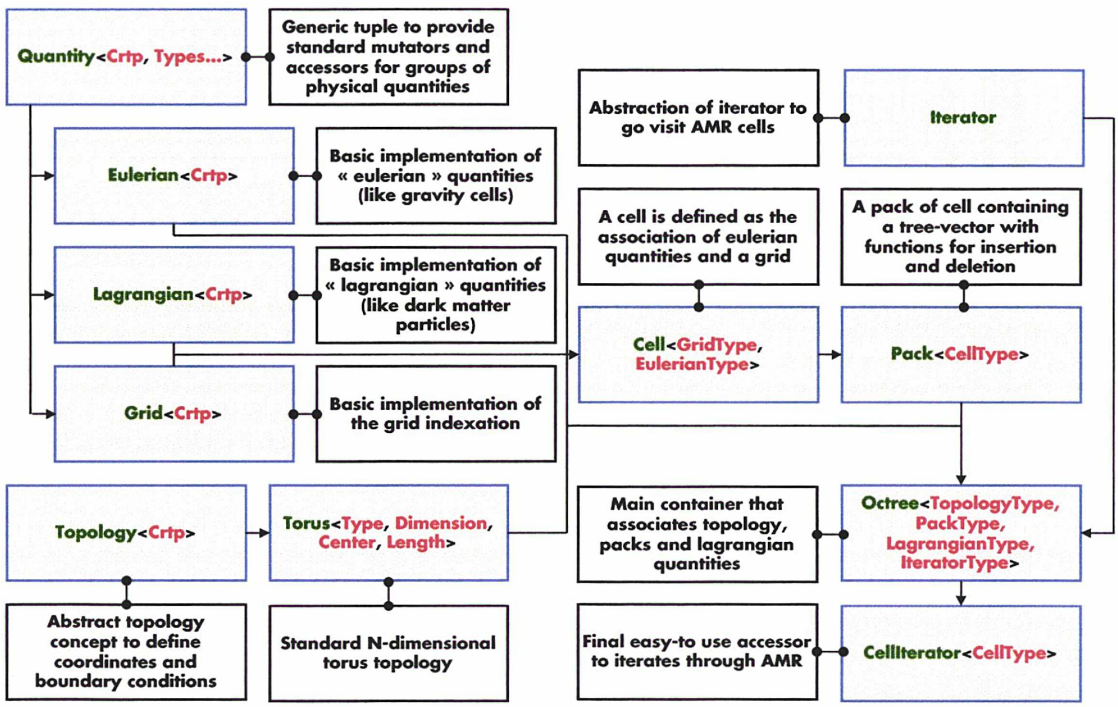


Figure 56 – Illustration d'une composante statique possible basée sur des classes template pour un code cosmologique. L'architecture présentée en figure 55 peut être déclinée de multiples façons différentes. Ci-dessus est présenté un diagramme de classes épuré envisageable pour la partie statique d'un code cosmologique. Les classes templates sont représentées en vert, les paramètres templates sont en rouge et les descriptions sont fournies en noir. On y retrouve des quantités eulériennes et lagrangiennes génériques utilisant le Curiously Recurring Template Pattern et dérivant d'une classe « tuple-like » `Quantity`. Une cellule est définie comme un ensemble formé d'une quantité eulérienne et d'informations concernant la grille telles que la position et le niveau. Les cellules sont elles-mêmes regroupées par « packs ». La topologie de la simulation, en l'occurrence un tore de dimension 3 est spécifié indépendamment et est injecté dans l'octree qui s'occupe de mailler l'espace. La navigation dans l'arbre s'effectue aux moyens d'itérateurs génériques.

Chapitre 6

Vers des orthtrees plus raffinés

***

6.1 Position du problème . . . 132
6.1.1 Description d'une implémentation classique : le cas de Ramses . . . 132
6.1.2 Limitations des implémentations traditionnelles . . . 133
6.2 Solution unifiée . . . 136
6.2.1 Principe de l'algorithme . . . 136
6.2.2 Propriétés locales et encodage des indices . . . 137
6.2.3 Propriétés globales et émergence de l'arbre . . . 145
6.2.4 Avantages et limitations . . . 148
6.3 Application et perspectives . . . 152
6.3.1 Implémentation et améliorations . . . 152
6.3.2 Vers une généralisation à tout type d'arbre . . . 153

Comme on l'a vu au chapitre précédent, il est possible de mettre en évidence une architecture logicielle commune capable de satisfaire la plupart des besoins en cosmologie numérique. Le premier niveau de cette architecture est la partie statique : celle qui consiste à repérer des données distribuées dans l'espace et naviguer au sein de ces données pour y effectuer des calculs tels que des moyennes de quantités dans des sous-volumes, de la détection de structures ou encore du calcul de fonctions de corrélation. Les arbres constituent la structure de données centrale de ce niveau : dans le cas des maillages adaptatifs pour des codes dynamiques, bien entendu, mais également par exemple dans le cas de l'analyse de données observationnelles à l'aide de kd-tree. L'objectif de ce chapitre est de proposer une nouvelle façon de résoudre la question de l'implémentation des orthtrees, généralisations des octrees [1] en dimension arbitraire. Le chapitre se concentre sur la description du cœur d'un nouvel algorithme pour la cosmologie numérique.

Comment organiser intelligemment des données physiques distribuées dans l'espace ? Voilà une question qui hante la plupart des concepteurs des codes de simulation les plus utilisés dans la communauté astrophysique. La question pourrait presque paraître anodine, et relever du détail d'implémentation. En réalité, elle est absolument centrale car d'elle dépend une grande partie de la structure des codes, la façon dont les équations vont pouvoir être résolues, la facilité avec laquelle la parallélisation pourra être mise en œuvre et les performances qui pourront être atteintes. De la même manière qu'un instrument scientifique conditionne les questions physiques qui peuvent être abordées grâce à lui, un

1. Arbre dont les nœuds peuvent compter jusqu'à 8 enfants utilisé pour des maillages adaptatifs cartésiens en trois dimensions.

élément aussi structurant que celui dont il est sujet ici limite les pistes théoriques qui peuvent être étudiées. Réduire la complexité algorithmique, diminuer l'empreinte mémoire, améliorer la parallélisation et maximiser l'efficacité du transfert des données dans le cache des processeurs sont autant de moyens pouvant conduire à l'implémentation et la résolution de problèmes physiques plus complexes. Cela revêt une importance toute particulière dans le contexte de futures simulations sur machines exaflopiques dotées d'une très grande capacité de calcul mais d'une quantité de mémoire par cœur très limitée et pour lesquelles la localité des données sera un élément capital pour assurer des performances maximales.

La question posée dans ce chapitre est en fait relativement naïve : est-il possible de concevoir un orthtree à la fois performant, économe en mémoire, efficace du point de vue du cache, respectant la localité des données, trivial à paralléliser et générique tout en préservant autant que faire se peut une simplicité d'implémentation et d'utilisation ? Nous allons voir que cela est non seulement possible, mais aussi aisément réalisable à partir d'un principe de base à la fois simple et puissant. Si certaines approches utilisées en géométrie computationnelle[132, 133] peuvent présenter des similitudes avec celle décrite dans la suite de ce chapitre, cette dernière n'a pour le moment jamais été mise en œuvre dans le cadre de la cosmologie numérique. Sous un œil de physicien, l'on pourrait dire qu'il ne va pas s'agir de se concentrer sur les concepts d'arbre et de space filling curve, mais plutôt de mettre au jour un objet plus fondamental, exhibant une dualité langage de programmation/nombre entier, duquel émergeront naturellement à la fois des propriétés d'arbre et de courbe de Lebesgue multiniveau. À l'issue de ce chapitre, la construction d'un arbre de raffinement adaptatif se résumera au tri d'un tableau d'indices entiers, à partir duquel toutes les bonnes propriétés énumérées dans la question formulée quelques lignes plus tôt émergeront.

6.1 Position du problème

6.1.1 Description d'une implémentation classique : le cas de Ramses

Le point de départ de cette réflexion est l'implémentation de l'arbre de raffinement adaptatif dans Ramses. Il s'agit ici d'un exemple parmi d'autres, les autres codes les plus répandus au sein de la communauté de la cosmologie numérique se basant sur des approches du même type. Pour distribuer les données dans l'espace, Ramses utilise deux éléments : un octree et une space filling curve, abréviée SFC dans la suite. Le premier est à la base du maillage : là où la densité de matière est plus élevée, les cellules se subdivisent récursivement en huit cellules filles jusqu'à atteindre soit un seuil de densité, soit un niveau maximal. Le second est lié à la maximisation de la localité des données du point de vue de la parallélisation. En l'occurrence c'est une courbe de Hilbert qui est utilisée.

Toutefois, en plus du problème du manque de généricité évoqué au chapitre précédent, l'implémentation actuelle présente des problèmes algorithmiques profonds qui ne feront qu'empirer à mesure que le code sera porté sur des machines disposant de plus en plus de cœurs. En effet celle-ci repose sur une approche classique en termes de pointeurs. Chaque élément de grille dispose ainsi au moins d'un indice, de l'indice du parent, de l'indice du premier fils s'il existe, des indices des grilles précédente et suivante pour la navigation dans l'arbre et des coordonnées normalisées du centre x, y, z entre 0 et 1. Il s'agit juste là du stockage de l'arbre : les données physiques (ρ, T, P, ...) contenues dans les éléments de grilles sont stockés séparément. Une représentation d'un tel arbre de maillage adaptatif est donné en figure 58. On notera que le choix de normaliser les coordonnées entre 0 et 1 présente un inconvénient du point de vue de la précision flottante : la précision n'est pas uniforme sur le domaine, avec un avantage pour les valeurs proches de 0. Il existe trois solutions pour pallier à ce problème : le symétriser en passant à l'intervalle $[-0.5, 0.5[$, l'uniformiser en passant à l'intervalle $[1, 2[$, ou passer à une représentation arithmétique à virgule fixe. Concernant les indices des grilles, il s'agit en l'occurrence de nombres entiers stockés dans des tableaux alloués sur chaque tâche MPI, mais cela est

équivalent à des pointeurs puisque la navigation dans l'arbre consiste à passer d'indice en indice. Les grilles sont groupées dans des tableaux par niveau de raffinement. L'arbre est « fully-threaded » en ce sens qu'il contient les indices des cellules voisines pour accélérer la navigation. Du point de vue de la parallélisation, les grilles sont rangées suivant la courbe de Hilbert ce qui permet de les répartir sur les processeurs en suivant l'ordre des indices.

La figure 57 illustre les différents points susmentionnés : au départ, les particules résultant de la discrétisation du champ de matière se déplacent au sein d'une grille grossière, sans raffinement. Puis, à mesure que la matière s'effondre, les niveaux de raffinements se déclenchent. Visuellement, cela correspond à subdiviser chaque cellule des régions surdenses en huit nouveaux éléments de maillage. Dans la pratique, cela correspond à référencer, pour chaque subdivision, huit nouveaux ensembles de données $i, x, y, z, \rho, \ldots$ et à créer les liens adéquats entre parents, enfants et voisins pour assurer la navigation au sein de l'arbre. Comme dit plus haut, l'équilibrage de la charge sur les différents processeurs est assurée par une courbe de Hilbert qui parcourt le maillage : sur la figure 57, chaque couleur correspond à une zone traitée par une tâche MPI. L'avantage de la courbe est double : d'une part cela minimise globalement la surface totale des sous-volumes traités par chacune des tâches, et d'autre part deux tâches consécutives traitent deux zones spatiales proches, facilitant ainsi potentiellement les communications au sein des supercalculateurs.

6.1.2 Limitations des implémentations traditionnelles

Si cela peut sembler, sur le papier, optimal et adapté à des simulations de plusieurs centaines de milliards d'éléments de grille, il y a, dans la pratique un certain nombre de problèmes. Tout d'abord, la courbe de Hilbert : elle introduit une complexité telle dans la distribution des données que des codes de post-traitement ont été créés pour regénérer des ensembles cubiques de grilles à partir des fichiers bruts fournis par Ramses. La courbe de Hilbet conduit donc mécaniquement à une parallélisation complexe ne serait-ce que pour réorganiser l'indexation des données lorsque des éléments de grille sont redistribués sur les différentes tâches MPI. Il ne s'agit pas là d'un problème algorithmique mais d'un problème purement pratique.

Au niveau des performances le premier problème vient de la mémoire. En effet, un très grand nombre de quantités est stocké pour chaque nœud de l'arbre : les coordonnées, les indices du parent, du premier enfant et des voisins, ainsi que des booléens pour stocker des informations tels que la nécessité de raffiner ou déraffiner la grille au pas de temps suivant. L'empreinte mémoire de la structure de l'arbre est donc loin d'être négligeable. Ensuite, la navigation pose un triple problème du point de vue du processeur. Le premier concerne la localité des données : en effet des grilles voisines n'ont aucune raison de l'être dans la mémoire tout comme des éléments de grilles parents et fils. Cela empêche de tirer parti du cache, puisque le processeur ne cesse de naviguer dans la mémoire. Il s'agit là d'un point crucial pour l'optimisation des codes sur les futures machines exascale, certains allant même jusqu'à dire que le temps de calcul sera « gratuit » en comparaison du temps nécessaire au déplacement des données. Naviguer de pointeurs en pointeurs, c'est donc prendre le risque de passer la plus grande partie de son temps à transférer des données entre la mémoire et le cache. Le second problème, qui accompagne le premier, est l'absence de prédictibilité de navigation dans la mémoire, ce qui empêche un prefetching [2] efficace. Il est en effet inutile de précharger les éléments contigus en mémoire puisqu'il est peu probable qu'il s'agisse des prochains éléments accédés. Enfin, le troisième problème concerne la vectorisation des opérations, qui est là encore empêchée du fait des indirections. Cela est d'autant plus problématique qu'il s'agit d'un élément central pour les performances du calcul sur GPU ou sur accélérateurs manycores. De ce fait, la grande majorité des portages sur GPU concernent des codes dynamiques sur grille fixe et non sur grille raffinée. Pour finir, la distribution des particules dans le

2. Technique d'optimisation de l'exécution des programmes consistant à précharger les instructions ou la mémoire de manière séquentielle en vue de réduire les temps d'attentes au niveau du processeur.

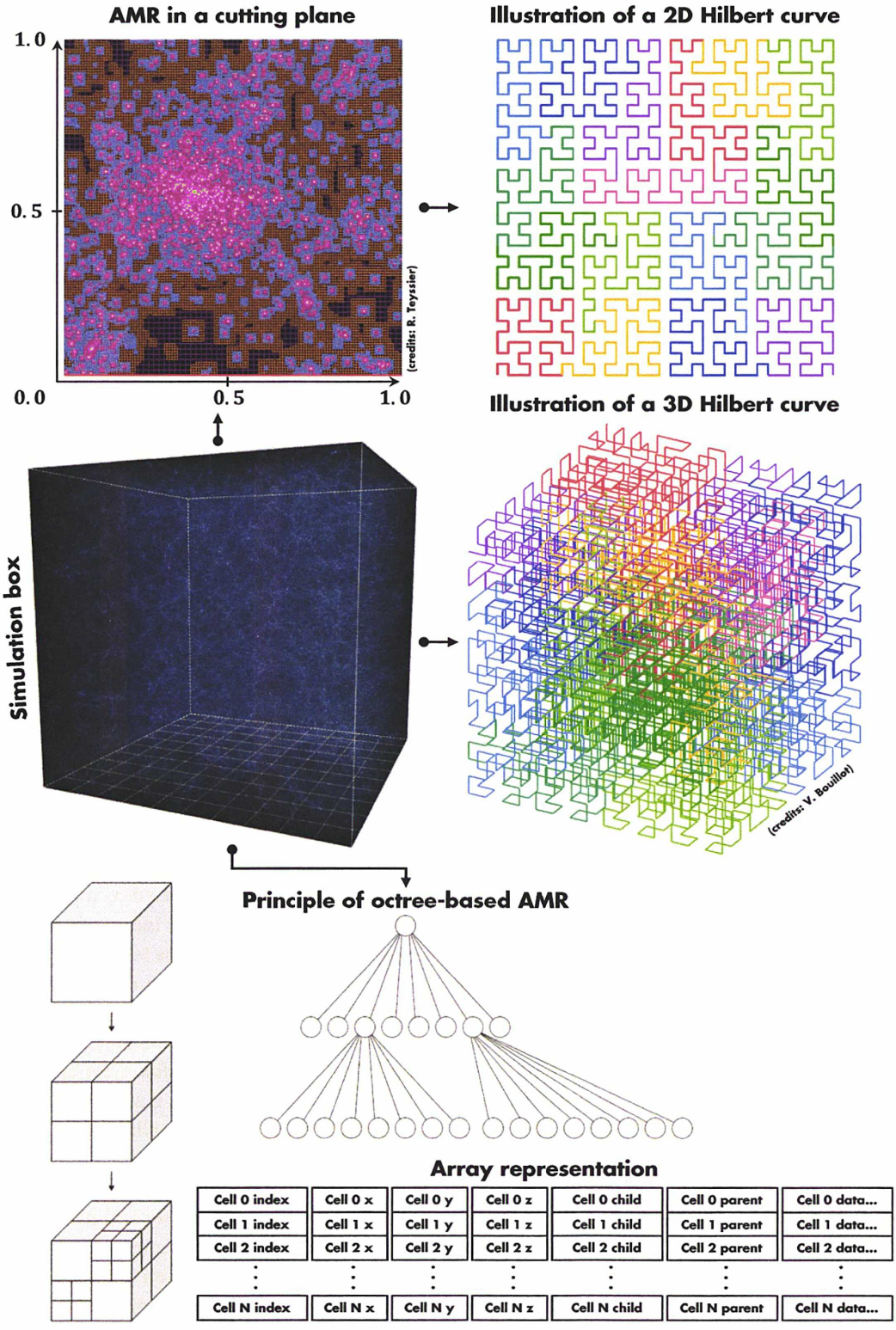


Figure 57 – Octree et courbe de Hilbert dans Ramses. La représentation du contenu d'un boîte de simulation en termes de densité de matière apparait au centre. Pour suivre l'effondrement gravitationnel, le code utilise une grille adaptative dont le comportement est illustré en haut à gauche : chaque cellule est divisée en huit dès lors qu'elle franchit un seuil de densité, ce qui permet d'accroître la résolution spatiale dans les zones surdenses. Le principe du maillage adaptatif basé sur un octree est donné dans la partie basse de la figure. Dans la pratique, un ensemble de tableaux contient toutes les données nécessaires pour chaque cellule, ainsi que les liens qui existent entre ces dernières. La répartition de la charge sur les différents processeurs est déterminée une courbe de Hilbert passant au centre des éléments de maillage et qui permet d'assurer la localité des données.

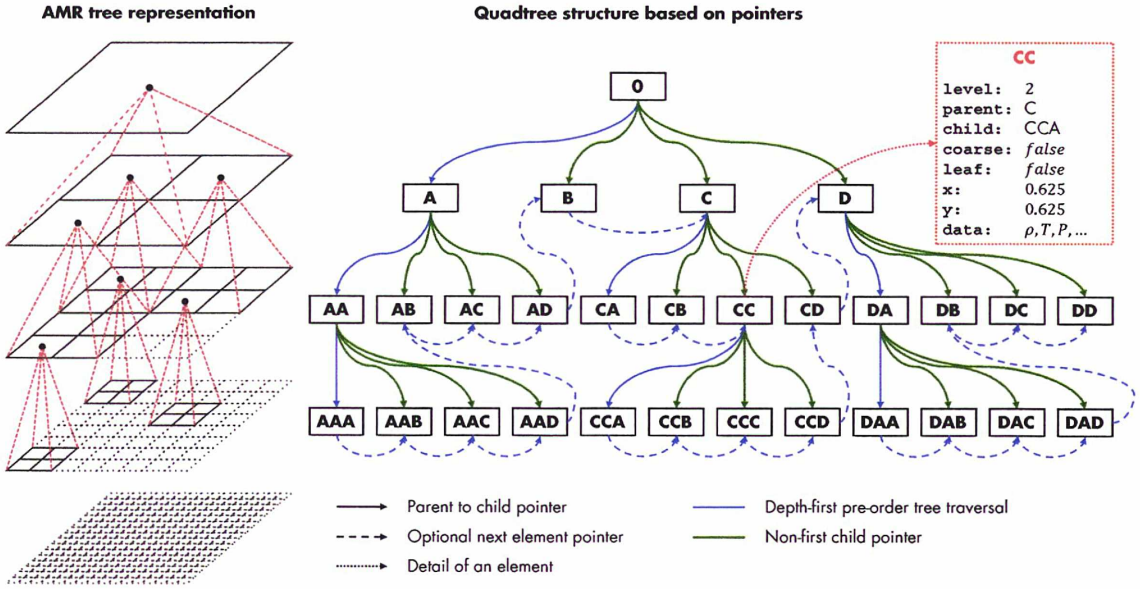


FIGURE 58 – Raffinement adaptatif basé sur un orthtree de dimension 2 et structure de données correspondante. On considère le maillage adaptatif basé sur un quadtree donné à gauche : chaque raffinement correspond à la division d'une cellule en quatre. Dans cet exemple, on a au plus quatre niveaux, mais les cellules feuilles les plus profondes sont ici au niveau 3. Ce genre de structure est ce qui peut permettre d'accroître la résolution dans des zones d'intérêt de simulations physiques et c'est ce qui est utilisé par certains codes comme RAMSES pour suivre l'effondrement de la matière dans les simulations cosmologiques. La représentation « naturelle » d'un tel maillage est celle en termes d'arbre donnée dans la partie droite de la figure : chaque cellule constitue un nœud de l'arbre qui peut soit ne pas avoir d'enfant dans le cadre d'une cellule feuille, soit en avoir 4 dans le cadre d'une cellule raffinée. Les lettres ne représentent ici rien de concret et sont juste utilisées pour différencier les cellules. La structure en arbre émerge du fait que les nœuds sont liés par des pointeurs, que ceux-ci correspondent à des pointeurs mémoire, ou que ceux-ci soient des liens au sein de tableaux comme dans le cas de RAMSES. Dans une telle structure de données, chaque nœud contient les données de la cellule associée comme illustré ici pour l'élément de maillage « CC » : en plus des liens vers la cellule parent et la première cellule fille le cas échéant, il est généralement nécessaire de stocker le niveau et la position de la cellule, de même que des informations concernant la structure du maillage et bien entendu, toutes les informations physiques. Le surcoût lié au maillage, c'est-à-dire l'empreinte mémoire représentée par tous les éléments autres que l'information physique est donc loin d'être négligeable. Et cette empreinte ne peut qu'augmenter si l'on cherche à ajouter des liens optionnels pour améliorer les performances de la navigation. Ici, on a représenté l'utilisation de pointeurs vers le prochain élément pour faciliter le parcours de l'ensemble de l'arbre dans le cadre d'une navigation en profondeur préfixée (« depth-first pre-order tree traversal »). Du point de vue des simulations cosmologiques, une telle représentation présente deux problèmes : le premier vient de l'empreinte mémoire importante de la structure de données liée au stockage de tous les liens, le second vient de la fragmentation des données en mémoire qui impacte significativement les performances du fait des indirections. Avec de telles structures, le processeur passe son temps à accéder à la mémoire principale pour naviguer de cellule en cellule, et cela pose un problème dans un contexte où le transfert des données entre la mémoire vive et les unités de calcul est susceptible de devenir un goulot d'étranglement des architectures exaflopiques.

maillage nécessite de maintenir à jour une liste d'indices pour savoir à quel élément de grille est associé chaque particule.

Toutes ces limitations proviennent du fait que le raffinement adaptatif est traduit en termes d'arbre classique, basé sur des pointeurs, comme illustré en figure 58. Ce sont ces pointeurs, pourtant apparemment nécessaires pour matérialiser les liens entre cellules parents, cellules filles et cellules voisines qui introduisent les indirections qui participent au fractionnement de la mémoire et empêchent nombre d'optimisations. Et même si l'ajout de liens optionnels peut accélérer la navigation, comme par exemple l'utilisation des pointeurs enfants des cellules feuilles comme liens vers les nœuds suivants du point de vue d'un parcours en profondeur préfixé [3], il ne résout ni le problème de l'occupation mémoire, ni celui

3. Appelé en anglais « depth-first pre-order tree traversal ».

du fractionnement de cette dernière, source de performances dégradées. D'ailleurs, en gardant la structure de données classique illustrée en figure 58, accélérer la navigation se traduit dans la très grande majorité des cas par une augmentation de l'empreinte mémoire. En effet, comme évoqué plus haut, l'amélioration des performances peut passer par l'ajout de liens supplémentaires tels que des liens vers tous les voisins, mais ces derniers entraînent un surcoût de mémoire, qui peut vite devenir prohibitif dans un contexte où la mémoire est l'un des aspects limitant les plus critiques des simulations cosmologiques. S'opère donc traditionnellement un équilibrage performances/mémoire sur lequel jouent les différents codes principaux utilisés dans la communauté, et les différentes versions de ces codes. Mais plutôt que de jouer sur cet équilibrage, l'on va se demander dans la suite s'il n'existerait pas une autre manière de représenter les données qui permettrait de solutionner naturellement l'ensemble des problèmes évoqué plus haut.

6.2 Solution unifiée

6.2.1 Principe de l'algorithme

Est-il possible d'apporter une solution unique à toutes les problématiques soulevées dans la section précédente ? C'est ce que l'on va tenter de montrer ici. Deux présentations seraient envisageables : l'une présentant par tâtonnements l'élaboration de la solution proposée, l'autre présentant d'abord la solution et montrant *a posteriori* en quoi elle permet de retrouver toutes les bonnes propriétés. C'est la seconde approche qui a été choisie, mais la simplicité de la solution proposée ne doit pas éclipser le fait qu'obtenir les caractéristiques souhaitées en termes de performances de calcul et d'empreinte mémoire n'est pas facile. D'autre part, le maillage étant l'une des structures les plus centrales des codes cosmologiques, améliorer les performances de navigation ou l'empreinte mémoire de ce dernier de quelques pourcents représente un gain appréciable pour l'ensemble des codes. Des benchmarks préliminaires semblent montrer que remplacer les arbres classiques par la structure présentée ici est à même d'assurer des gains largement au-delà de quelques pourcents ou dizaines de pourcents sur ces deux aspects.

La solution, relativement simple, réside dans l'utilisation d'arbres implicites tirant parti du fait que le nombre maximal de fils pour chaque nœud de l'arbre est connu à la compilation $N_{\text{children}} = 2^{N_{\text{dimension}}}$, de même que le niveau maximal de raffinement N_{level}. Toute la structure de l'arbre peut alors se résumer à un entier par nœud. La détermination de cet indice i peut être vu comme une suite d'instructions, encodée au niveau binaire. Ce mini-langage, comparable, en un sens, à de l'assembleur est le suivant. Il comprend deux opcodes :

- `stop` encodé par 0
- `goto` encodé par 1

et $2^{N_{\text{dimension}}}$ opérandes associées aux directions possibles, c'est-à-dire par exemple en 2 dimensions :

- `lower left` encodé par 00
- `lower right` encodé par 01
- `upper left` encodé par 10
- `upper right` encodé par 11

et repose sur les règles suivantes :

- chaque instruction est composée exactement d'un opcode suivi d'une opérande
- le premier `stop` rencontré termine le programme
- le premier `stop` rencontré ne peut être suivi que d'opcodes et d'opérandes nuls

À partir de ces règles, on peut écrire des programmes du type :

```
goto lower right
goto lower left
goto upper left
stop
```

qui permettent de décrire un itinéraire de navigation dans un orthtree, les `goto` correspondant à une descente à un niveau plus raffiné et les `stop` permettant de s'arrêter à la cellule recherchée. Ainsi ce programme permet d'accéder à un élément de grille du quatrième niveau en empruntant les directions indiquées pour chaque niveau. Conformément aux règles d'encodage énoncées plus tôt, cela s'écrit en binaire :

```
1 01
1 00
1 10
0 00
0 00
0
```

où toutes les instructions suivant le premier `stop` ont été mises à zéro conformément aux règles énoncées précédemment. Cet ensemble de commandes peut aussi se réinterpéter comme l'encodage d'un nombre entier de N_{bits}, en prenant la convention suivante :

- le programme commence par le bit de poids le plus fort

ce qui donne en l'occurrence $1011001100000000_2 = 45824_{10}$. D'autres exemples sont illustrés en figure 59. En attribuant à chaque élément de grille un indice entier non signé sous forme de mini-programme et en triant le tableau d'indices ainsi obtenu, on peut montrer que l'on obtient un orthtree disposant de toutes les propriétés recherchées. Autrement dit, remplacer un arbre classique par un tableau de paires clé-valeur où les clés correspondent à des mini-programmes et où les valeurs correspondent aux données physiques des cellules, et classer ce tableau par clés croissantes apporte une solution à l'ensemble des problèmes évoqués précédemment. Les paragraphes suivants montrent comment et pourquoi la mise à profit d'une telle dualité mini-programme/nombre entier pour le calcul d'indices résout ces problèmes.

6.2.2 Propriétés locales et encodage des indices

Décodage de l'indice

On s'intéresse dans cette sous-partie aux propriétés locales, c'est-à-dire à toutes les caractéristiques qui concernent un indice isolé. Autrement dit comment extraire et encoder de l'information en utilisant les règles présentées dans la sous-section précédente.

Disposant d'un indice i, l'une des premières questions qui se pose est celle du repérage de la cellule associée. On l'a vu, chaque cellule peut être repérée de manière unique par le chemin suivi dans l'arbre : c'est justement ce qui a permis de déterminer l'indice i. Mais chaque cellule peut aussi être repérée de manière unique par le niveau auquel elle appartient ainsi que par ses coordonnées le long de chacune des dimensions. Le calcul efficace de ces quantités est essentiel pour se situer dans l'espace : il s'agit d'opérations de décodage de l'indice qui seront appelées en permanence dans le cadre de simulations cosmologiques. Une façon possible de procéder est présentée en figure 60 ci-dessous.
Comme illustré dans l'exemple, le niveau peut être obtenu simplement en commençant par masquer les opcodes de l'indice à 1 puis en comptant le nombre de bits mis à 1 dans le résultat ou, dit autrement, en effectuant : `level = popcnt(index & mask)`. L'extraction de la coordonnée suivant chaque dimension au niveau de la cellule est simplement obtenue via un « parallel bits deposit » en utilisant le masque approprié : `coord = pdep(index & mask)`. Dans l'exemple précédent la cellule est située au niveau 3, et à `x = 7`, `y = 4` et `z = 2`[4]. Ou encore, si l'on souhaite obtenir les coordonnées normalisées

4. L'ordre des coordonnées aurait pu être choisi différemment.

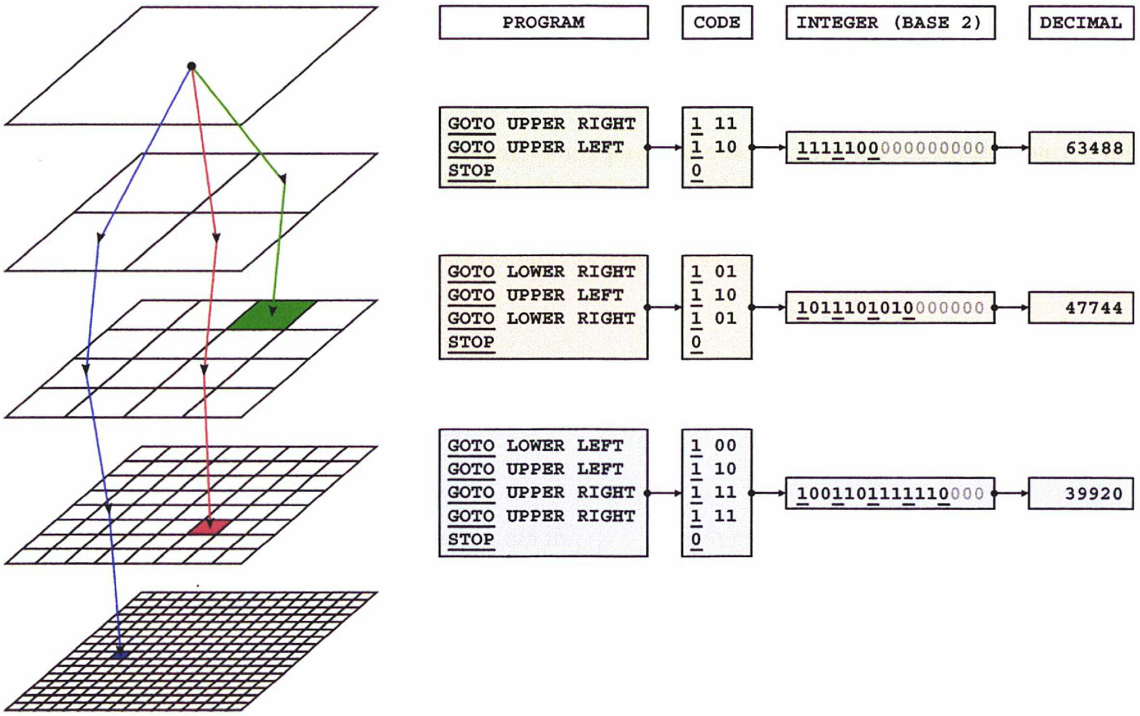


Figure 59 – Indexation de cellules et dualité mini-programme/nombre entier. On considère trois cellules dans un orthtree de dimension deux et l'on cherche leur encodage du point de vue de l'algorithme présenté dans cette partie. Chacune de ces cellules peut être repérée de manière unique par le chemin suivi dans l'arbre depuis la racine du niveau 0. Ce chemin est ensuite traduit en mini-programme. À chaque mini-programme correspond un code binaire selon les règles énoncées précédemment. Ce code binaire peut ensuite être stocké dans un entier en commençant par le bit de poids le plus fort et en complétant par des 0 : les opcodes sont ici soulignés, les opérandes apparaissent en caractères normaux, et les zéros de complément sont grisés. Ici, on considère des entiers de 16 bits, mais rien n'empêcherait de considérer des entiers de taille supérieure. Comme on va le montrer dans la suite, la clé de l'algorithme réside dans le fait que ces représentations binaires peuvent se réinterpréter comme des nombres entiers, qui, une fois stockés dans un tableau et classés par ordre croissant font émerger toutes les bonnes propriétés.

à 1, il s'agit de la cellule centrée en :

$$(x,y,z) = \left(\frac{7\times 2+1}{2^{3+1}}, \frac{4\times 2+1}{2^{3+1}}, \frac{2\times 2+1}{2^{3+1}}\right) = (0.9375, 0.5625, 0.3125) \tag{6.1}$$

On remarque d'ailleurs que, contrairement au cas de Ramses, les coordonnées sont ici stockées sous forme de valeur entière, c'est-à-dire sans approximation et sans le problème des virgules flottantes évoquées dans la première section de ce chapitre. Comme on le voit, l'encodage de l'indice permet donc facilement de retrouver la position de la cellule associée dans l'arbre.

Performances des manipulations de bits

Ces premiers calculs sont l'occasion de faire plusieurs remarques du point de vue de l'empreinte mémoire et des performances. Contrairement au cas classique où les informations sur la position de la cellule sont stockées en clair, elles sont ici toutes encodées dans l'indice. Cela peut se traduire par un gain d'empreinte mémoire potentiellement important puisque l'on peut par exemple passer de $4 + 3 \times 4 = 16$ octets pour le stockage d'un entier et 3 flottants simple précision à un seul entier de 64 bits pour un indice de 8 octets. C'est-à-dire un facteur 2 en termes de mémoire. Pour l'instant cela ne concerne pas les pointeurs, cela concerne uniquement la localisation de chaque cellule. Cependant, comme les données n'apparaissent plus en clair, il est nécessaire de les recalculer à chaque fois. Autrement dit, le gain mémoire a été troqué contre du temps de calcul. La pertinence de l'algorithme débattu ici repose donc notamment sur les performances du décryptage : comme le niveau et

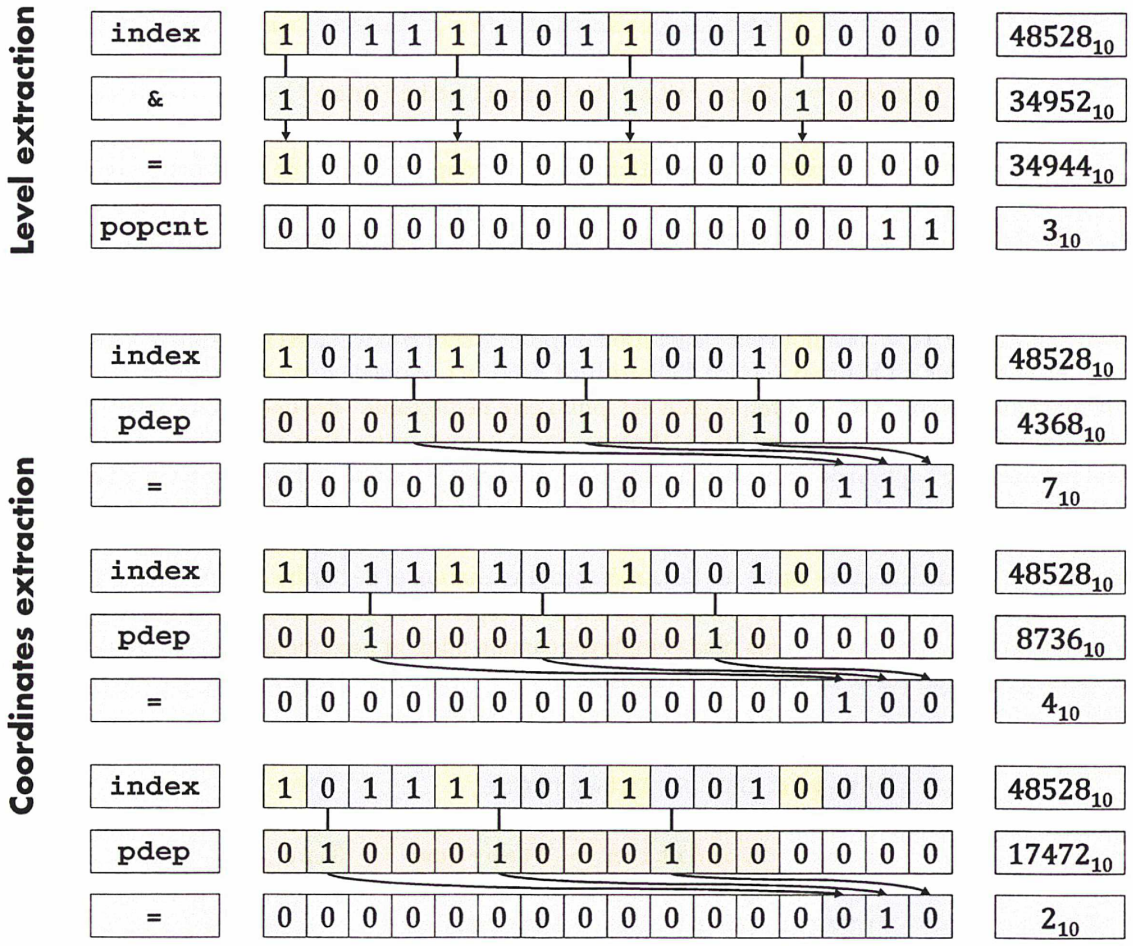


FIGURE 60 – Extraction du niveau et des coordonnées pour $N_{\text{dimensions}} = 3$, $N_{\text{bits}} = 16$ et pour l'indice de cellule $i = 48528$. Les opcodes apparaissent en jaune, les opérandes en bleus, et les masques utilisés sont présentés en rouge et vert. L'extraction du niveau repose sur un « population count » tandis que l'extraction des coordonnées repose sur des « parallel bits deposit ».

les coordonnées doivent potentiellement être recalculés à chaque accès de cellule, si leur extraction est complexe, l'impact sur le temps de calcul pourrait être prohibitif. Heureusement ce n'est pas le cas, et c'est même plutôt l'inverse. En effet, les opérations présentées en figure 60 sont rapides, et ce, pour deux raisons. La première est que les masques ne dépendent que du nombre de dimensions $N_{\text{dimensions}}$ et de la taille des indices N_{bits}. Ces deux quantités étant déterminées à la compilation, on peut utiliser la métaprogrammation sur les valeurs pour précalculer tous les masques et laisser le compilateur procéder à tous les précalculs et optimisations qui en résultent. Par exemple, le masque associé aux niveaux peut-être obtenu récursivement par :

```
1  template <class Integer,
2            Integer Index = Integer(),
3            Integer Zero = Integer(),
4            Integer One = Integer(1),
5            Integer Size = sizeof(Integer)*CHAR_BIT,
6            Integer Condition = Index+One <= Size>
7  constexpr Integer opcode_mask(const Integer ndim)
8  {
9    return (ndim < Size)*(
10     (Condition ? (One << Index)*(Index%(ndim+One) == Zero)
```

```
11                  | opcode_mask<Integer, Index+Condition>(ndim)
12                  : Condition)
13      << (Index == Zero)*(ndim+Size%(ndim+One)));
14  }
```

La seconde raison est l'utilisation des opérations popcnt, pdep, mais aussi de pext pour « parallel bits extract », lzcnt pour « leading zeros count », tzcnt pour « trailing zeros count »..., ces dernières fonctions étant utiles dans le cadre de conversions vers et à partir d'indices de space filling curves. Toutes ces opérations sur la manipulation de bits font parties de nouveaux jeux d'instructions fournis par AMD et Intel pour améliorer, entre autres choses, les performances des applications cryptographiques : ABM [5], BMI1 [6], BMI2 [7] et TBM [8]. La nouvelle génération de processeurs Haswell est la première à disposer de BMI1 et BMI2. Par un heureux hasard, l'algorithme décrit ici se révèle donc être parfaitement adapté aux machines à venir. Car en effet, le gain est loin d'être négligeable : des opérations potentiellement complexes à programmer peuvent être réalisées via une unique instruction assembleur. Ainsi, en activant par exemple l'option de compilation -mpopcnt sous g++ pour une architecture Sandy Bridge, on obtient des gains pour le calcul du niveau de l'ordre de 10x comparativement à la fonction optimisée de la bibliothèque standard std::bitset::count, de l'ordre de 15x comparativement à une solution indépendante de l'architecture basée sur de la métaprogrammation et de l'ordre de 40x comparativement à un comptage récursif trivial des bits mis à 1. On peut espérer des gains du même ordre de grandeur, voire mieux, pour des instructions plus complexes comme pdep et pext pour l'instant implémentées de façon générique via de la métaprogrammation :

```
1  template <class Integer, Integer Mask,
2            Integer Step = Integer(), Integer Shift = Integer(),
3            Integer One = Integer(1),
4            Integer Size = sizeof(Integer)*CHAR_BIT,
5            Integer Condition = Step+One <= Size,
6            bool Temporary = (Mask >> Step) & Condition>
7  constexpr Integer pdep(const Integer n)
8  {
9    return Condition ? !Temporary
10     ? pdep<Integer, Mask, Step+Condition, Shift+Temporary>(n)
11     : (n >> Shift & Condition) << Step
12     | pdep<Integer, Mask, Step+Condition, Shift+Temporary>(n)
13     : Condition;
14 }
15
16 template <class Integer, Integer Mask,
17           Integer Step = Integer(), Integer Shift = Integer(),
18           Integer One = Integer(1),
19           Integer Size = sizeof(Integer)*CHAR_BIT,
20           Integer Condition = Step+One <= Size,
21           bool Temporary = (Mask >> Step) & Condition>
22 constexpr Integer pext(const Integer n)
23 {
24   return Condition ? !Temporary
25     ? pext<Integer, Mask, Step+Condition, Shift+Temporary>(n)
26     : (n >> Step & Condition) << Shift
27     | pext<Integer, Mask, Step+Condition, Shift+Temporary>(n)
```

5. Advanced Bit Manipulation.
6. Bit Manipulation Instruction Set 1.
7. Bit Manipulation Instruction Set 2.
8. Trailing Bit Manipulation.

```
28       : Condition;
29 }
```

Comme on peut le voir pour l'extraction des coordonnées, pdep dépose les bits de la valeur fournie là où le masque vaut 1[134, 135]. L'instruction réciproque pext extrait les bits d'une valeur pour les insérer aux positions où le masque vaut 1. Le fait que ces instructions soient gérées nativement par les nouvelles générations de processeurs joue un rôle central pour les performances de l'algorithme : le gain substantiel en mémoire a un coût quasi-nul en termes de performances puisque le décodage des indices fait appel à une ou deux instructions assembleur.

Calcul des indices des parents, enfants et voisins

Outre le fait d'être à même de calculer rapidement les coordonnées de chaque cellule dans l'arbre en fonction de son indice, il est important de pouvoir calculer efficacement les indices des parents, enfants, et voisins. Là encore, il s'agit d'opérations très simples à effectuer, reposant sur des opérations bit à bit natives, l'opération la plus complexe étant une fusion de deux valeurs en fonction d'un masque et pouvant s'écrire comme :

```
1 template <typename T, class = typename std::enable_if<
2           std::is_integral<T>::value>::type>
3 constexpr T merge(const T& a, const T& b, const T& mask)
4 {
5   return a ^ ((a ^ b) & mask);
6 }
```

Le détail des opérations de navigation d'indice en indice est donné dans la figure 61 ci-dessous.
Au final, pour chaque opération de déduction d'indice, seules quelques instructions assembleur sont nécessaires. Ce à quoi il faut bien entendu ajouter le temps de recherche et d'accès à la cellule en question dont il sera question dans la sous-section suivante. Mais le temps de calcul d'indice seul, lui, est très faible.

Indices et space filling curves

Á ces opérations de détermination de niveau, de position, de parents, d'enfants et de voisins, on peut ajouter un dernier type d'opération : celui de conversion vers des indices de space filling curve. En effet, comme nous allons le voir, la conversion des indices de cellules vers des indices monodimensionnels ou vers des indices de courbes de Lebesgue est relativement aisée et peut être utilisée encore une fois pour naviguer dans l'espace. La figure 62 présente quelques SFCs classiques : une courbe de remplissage « linéaire », une courbe de Lebesgue aussi appelée « z-curve » et la courbe de Hilbert utilisée dans Ramses. Les indices monodimensionnels « linéaires » sont les plus employés quand il s'agit de représenter un tableau multidimensionnel en mémoire. En trois dimensions, pour un tableau de taille $n_x \times n_y \times n_z$, et pour une cellule positionnée à (i_x, i_y, i_z) l'obtention de l'indice s'effectue par :

$$i_{\text{lin}} = i_x + i_y \times n_x + i_z \times n_x \times n_y \tag{6.2}$$

ou par :

$$i_{\text{lin}} = i_z + i_y \times n_z + i_x \times n_y \times n_z \tag{6.3}$$

suivant l'ordre dans lequel on arrange les dimensions. Dans le premier cas, les projections sur chaque axe peuvent être obtenues par :

$$i_x = i_{\text{lin}} \mod n_x \quad \text{et} \quad i_y = \left\lfloor \frac{i_{\text{lin}} \mod n_x \times n_y}{n_x} \right\rfloor \quad \text{et} \quad i_z = \left\lfloor \frac{i_{\text{lin}}}{n_x \times n_y} \right\rfloor \tag{6.4}$$

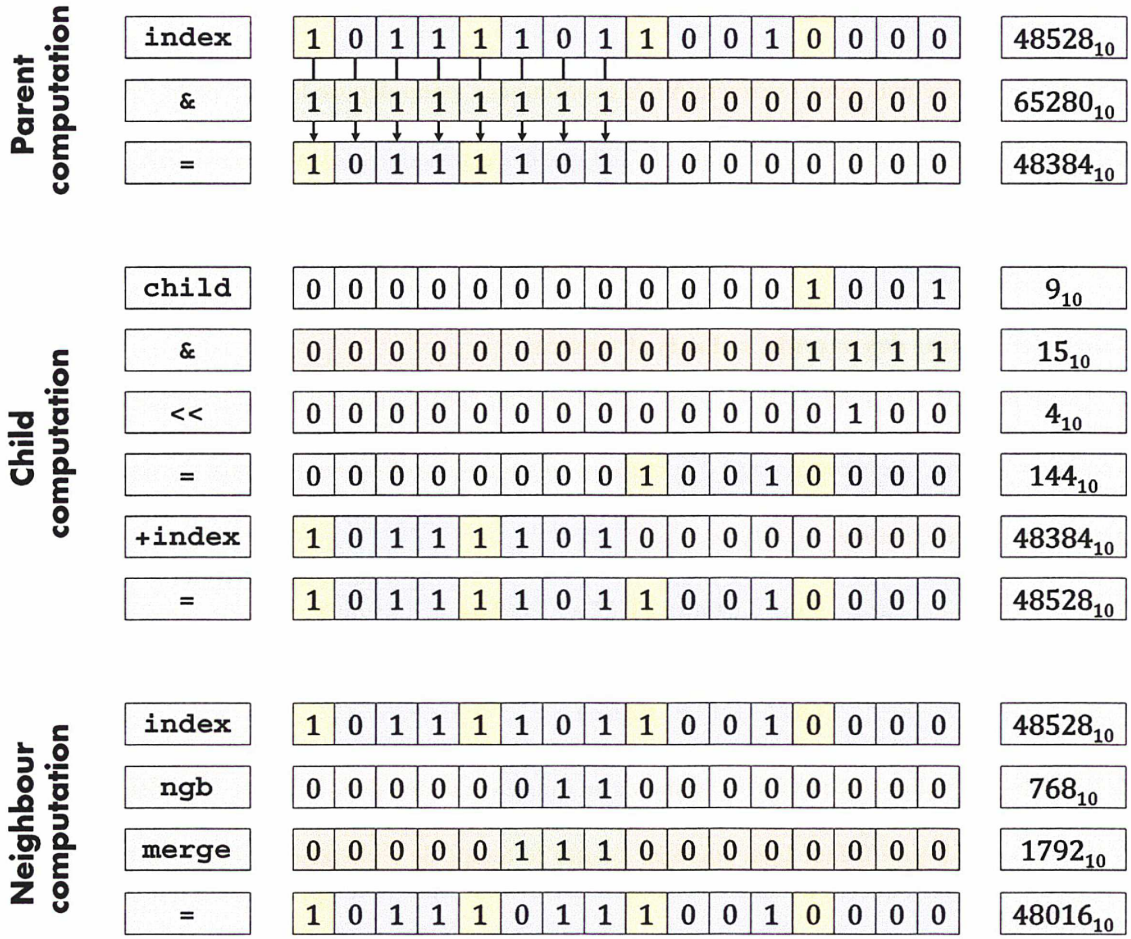


FIGURE 61 – Calcul des indices parents, enfants et voisins pour $N_{\text{dimensions}} = 3$ et $N_{\text{bits}} = 16$. Les opcodes apparaissent en jaune, les opérandes en bleus, et les masques utilisés sont présentés en rouge et vert. Les opérations impliquées sont des opérations bit à bit natives.

Il est possible d'obtenir directement l'indice linéaire à partir de l'indice déduit de l'algorithme présenté dans ces pages et inversement. La procédure à suivre est illustrée en figure 63. Elle repose sur des opérations d'entrelacement et de désentrelacement des bits un peu plus complexe que les opérations des paragraphes précédents, mais qui peuvent être programmées et optimisées en utilisant, encore une fois, des techniques de métaprogrammation. La figure présente également l'opération réciproque à savoir le calcul de l'indice de cellule à partir d'un indice monodimensionnel. L'intérêt de cette opération, s'il n'apparait pas immédiatement, est cependant essentiel. En effet, c'est grâce à elle, qu'il est possible de calculer efficacement l'indice d'une cellule connaissant sa position suivant les $N_{\text{dimensions}}$ et son niveau de raffinement. Pour cela, il suffit de calculer son indice monodimensionnel linéaire en utilisant l'équation 6.2 puis de convertir cet indice en utilisant le désentrelacement des bits. Il est également possible, et cela sera très utile dans la suite, de calculer l'indice de la cellule la plus raffinée possible à un endroit connaissant la position d'un point dans l'espace réel normalisé à 1. Dit autrement, pour tout x, y, z d'une boîte de simulation tridimensionnelle, il est possible de calculer facilement l'indice de la cellule la plus raffinée possible à cet position. Dans tous les cas, il s'agit de l'opération la plus coûteuse en termes de cycles d'horloges. Elle reste néanmoins limitée à quelques multiplications additions et opérations bit à bit si la fonction de désentrelacement a bien été optimisée à la compilation. On notera au passage qu'il est quelquefois plus rapide de générer et d'optimiser au maximum les opérations d'entrelacement et de désentrelacement pour tous les niveaux par métaprogrammation récursive et de brancher vers la bonne fonction à l'exécution, que de tout faire à l'exécution. Bien entendu, la mise à disposition des

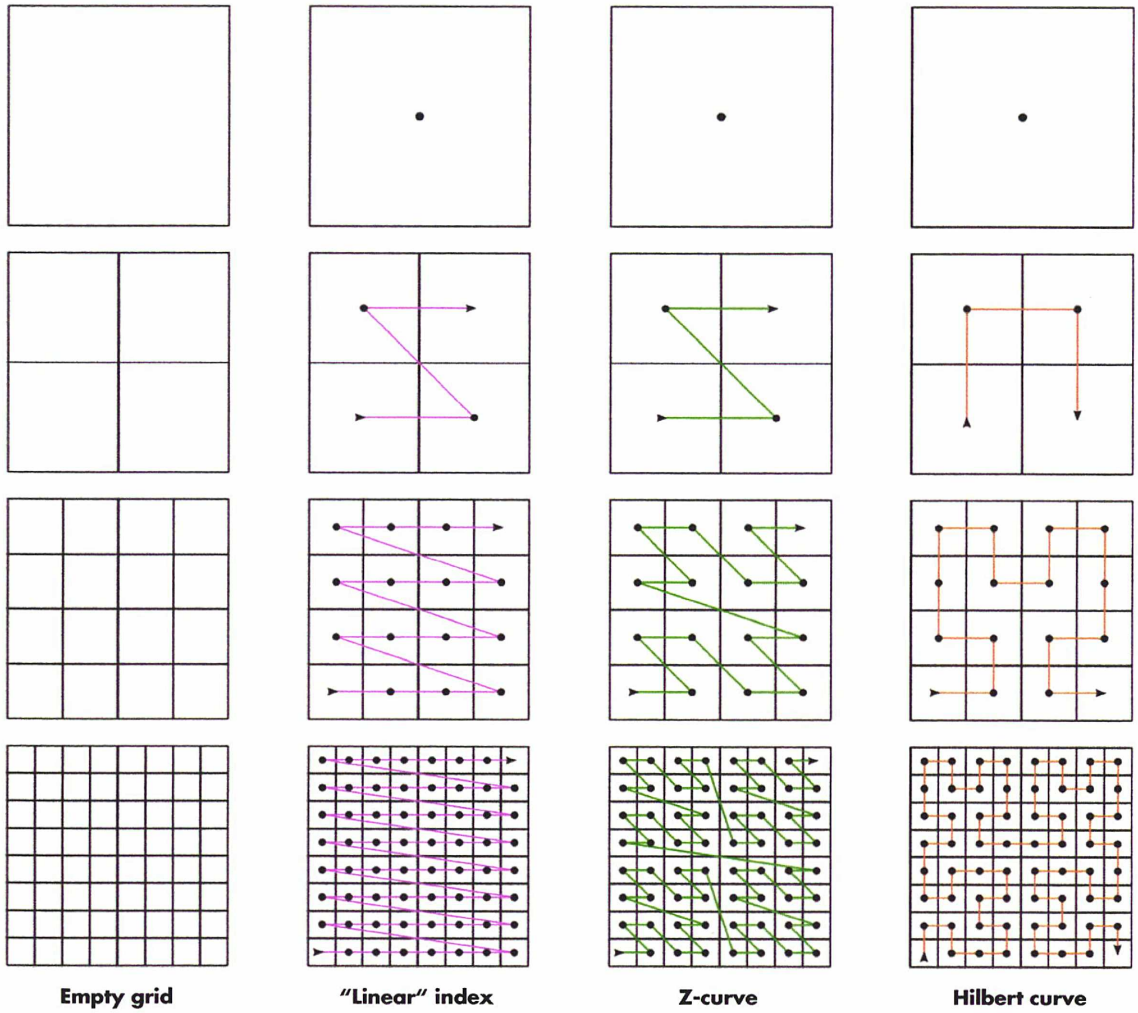


FIGURE 62 – Quelques exemples d'indexation de cellules selon des space filling curves. L'indice monodimensionnel « linéaire » est celui qui est le plus couramment utilisé sur des grilles fixes pour des simulations non parallélisées : il se calcule en trois dimensions simplement par $i_{\text{lin}} = i_x + i_y \times n_x + i_z \times n_x \times n_y$ où n_x, n_y et n_z représentent le nombre total de cellules suivant chaque dimension et où i_x, i_y et i_z représentent les coordonnées de la cellule le long de chaque axe. La courbe de Lebesgue, ou « z-curve » permet, elle, d'assurer davantage de localité aux données. Enfin, la courbe de Hilbert, utilisée dans RAMSES, permet d'assurer une localité encore plus grande. Elle peut ainsi conduire à des schémas de communication optimisés pour les supercalculateurs. Même si la courbe de Lebesgue parait moins optimale, elle présente d'autres avantages qui seront mis à profit dans la suite.

telles fonctions dans les jeux d'instructions des processeurs règlerait définitivement la question comme cela peut être le cas des opérations précédentes.

La dernière fonctionnalité importante réside dans la projection de l'indice sur la courbe de Lebesgue du niveau de la cellule. Cette fonctionnalité repose sur des « parallel bits deposit » et des « parallel bits extract ». Á partir de l'architecture HASWELL ces opérations peuvent être effectuées nativement par le processeur : leur exécution prend ainsi tout au plus quelques cycles d'horloge. En quelques cycles il est donc possible de retrouver le « z-index » d'une cellule à partir de son indice, et réciproquement. Il s'agit juste en fait d'ignorer les opcodes encodant le niveau, ou au contraire des les ajouter. La procédure est illustrée en figure 64. Dans le cas de la conversion depuis un « z-index », le masque impliqué est simplement calculé à partir d'un masque d'opcodes `opcode_mask` déterminé à la compilation, puis décalé à gauche en fonction du niveau de la cellule.

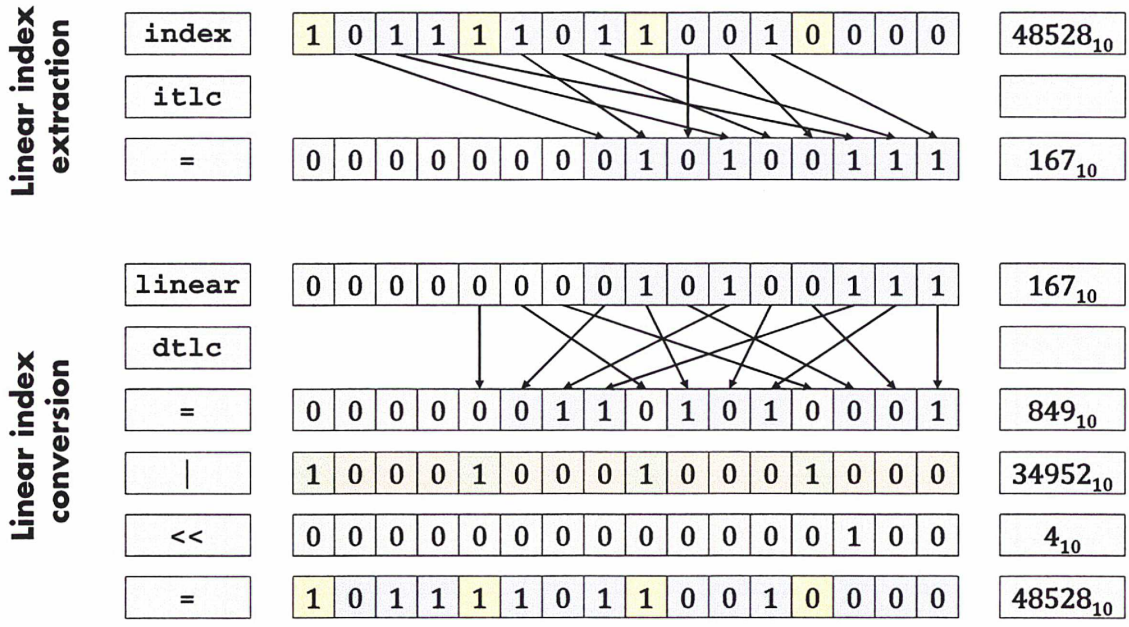


FIGURE 63 – Calcul de l'indice monodimensionnel linéaire pour $N_{\text{dimensions}} = 3$, $N_{\text{bits}} = 16$ et pour l'indice de cellule $i = 48528$. Les opcodes apparaissent en jaune, les opérandes en bleus, et les masques utilisés sont présentés en rouge et vert. L'extraction de l'indice monodimensionnel est effectuée via une opération d'entrelacement des bits, tandis que le calcul de l'indice de cellules à partir d'un indice monodimensionnel repose sur une opération de désentrelacement.

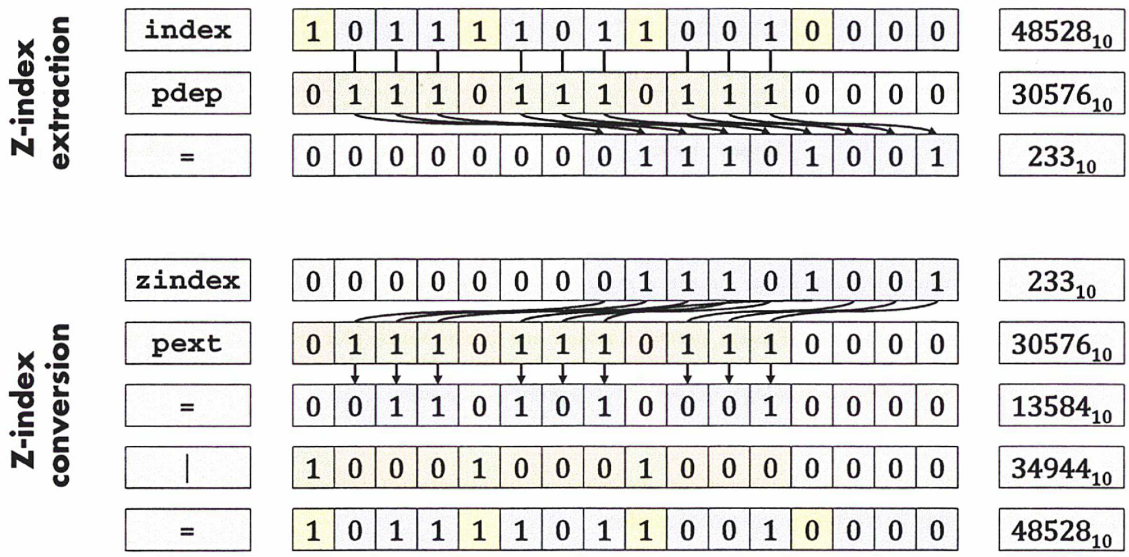


FIGURE 64 – Calcul de l'indice de la courbe de Lebesgue pour $N_{\text{dimensions}} = 3$, $N_{\text{bits}} = 16$ et pour l'indice de cellule $i = 48528$. Les opcodes apparaissent en jaune, les opérandes en bleus, et les masques utilisés sont présentés en rouge et vert. L'extraction de l'indice de Lebesgue repose sur un « parallel bits deposit » tandis que la réciproque repose sur un « parallel bits extract ».

La revue des opérations locales les plus utiles sur les indices s'arrête là. En quelques pages, nous avons montré comment extraire et encoder de l'information dans des indices à partir de l'algorithme décrit dans la première sous-section. D'autres opérations pourraient être programmées, mais celles présentées permettent déjà une grande liberté de navigation d'indice en indice. S'il s'agissait là d'explorer la partie « mini-programme », la sous-section suivante explore la réinterprétation de ces derniers en

termes de « nombres entiers » desquels émergent la structure en arbre et toutes les bonnes propriétés recherchées.

6.2.3 Propriétés globales et émergence de l'arbre

Caractère implicite de l'arbre et de la space filling curve

On s'intéresse maintenant aux propriétés globales, c'est-à-dire à toutes les caractéristiques qui concernent les liens qu'entretiennent les indices les uns avec les autres. Nous allons montrer comment la structure en arbre émerge naturellement de la réinterprétation entière des mini-programmes présentés dans la sous-section précédente.

L'obtention des bonnes propriétés est extrêmement simple : il suffit en effet de calculer les indices de toutes les cellules de l'arbre que l'on cherche à construire, de ranger ces indices dans un tableau, et de trier ce dernier par valeurs croissantes. Tout émerge alors naturellement du tableau : à la fois la structure en arbre, l'ordre de parcours en profondeur préfixé et une courbe de Lebesgue creuse multiniveaux dont le principe est illustré en figure 65. Pour le constater, le mieux est de procéder par l'exemple donné en figure 66. Comme on peut le voir sur cette figure, le tri du tableau des mini-programmes par valeurs entières croissantes donne naissance à un arbre implicite, rangé dans l'ordre de parcours en profondeur classique. Tout le contenu de la simulation peut alors se résumer à deux tableaux triés dans le même ordre, puisque qu'il est plus performant de faire appel à des structures de tableaux qu'à des tableaux de structures : un tableau de clés constitué par les indices et un tableau de données physiques. Quelques propriétés sont d'ores et déjà observables. Tout d'abord, l'arbre est implicite et ne nécessite aucun pointeur : il est donc extrêmement économe en mémoire puisque la navigation en son sein passe par une navigation dans un tableau. Ensuite, les grilles contigües sont relativement proches en mémoire, notamment en ce qui concerne les parents et les enfants : il n'y a pas de problème de fragmentation de la mémoire. Les clés sont triées selon un ordre de parcours en profondeur très classique, ce qui est un très bon signe pour le cache et le prefetcher. Enfin, l'ordre du tableau reproduit une space filling curve multiniveaux ce qui est un point positif pour ce qui est de la gestion de la parallélisation.

Navigation et recherche d'une cellule

L'ordre dans lequel sont rangés les éléments correspond à une courbe de Lebesgue multiniveaux passant, non pas seulement par les feuilles, mais aussi par tous les niveaux intermédiaires. Comme cela a déjà été évoqué, le simple parcours du tableau linéairement correspond à un algorithme de parcours en profondeur en ordre préfixé. Pour savoir si une grille est raffinée, il suffit de regarder le niveau de la grille suivante du tableau puisque chaque grille est directement suivie de la première grille fille si elle existe. Pour remonter à la grille parente il suffit de remonter le tableau en ordre inverse jusqu'à obtenir le premier indice de niveau inférieur. L'arbre peut être creux : la cohérence de celui-ci n'est pas remise en cause si le nombre de cellules filles est inférieur à $2^{N_{\text{dimensions}}}$. Le nombre de fils peut être connu en comptant le nombre d'indices de niveau $N_{\text{level}} + 1$ jusqu'à l'indice de niveau N_{level} suivant. Comme on peut le voir, la navigation dans l'arbre est très aisée puisque la structure d'arbre émerge naturellement du tri des indices, tri pouvant être effectué par exemple via un introsort[136], implémenté dans la librairie standard du C++ et déjà largement optimisée.

La recherche d'un élément est elle aussi très simple, puisque connaissant l'indice de l'élément recherché, il suffit d'effectuer une recherche binaire dans le tableau en $\mathcal{O}(\log N)$ où N représente la taille de ce dernier. Ce type de recherche étant très commun, il peut être optimisé au niveau assembleur par les compilateurs. Ainsi, la nature implicite de l'arbre converti une recherche par navigation dans l'arbre via des pointeurs, en une recherche d'entier dans un tableau trié, avec des mouvements en mémoire plus prédictifs puisqu'il est possible d'inférer la nature exacte des branchements plusieurs

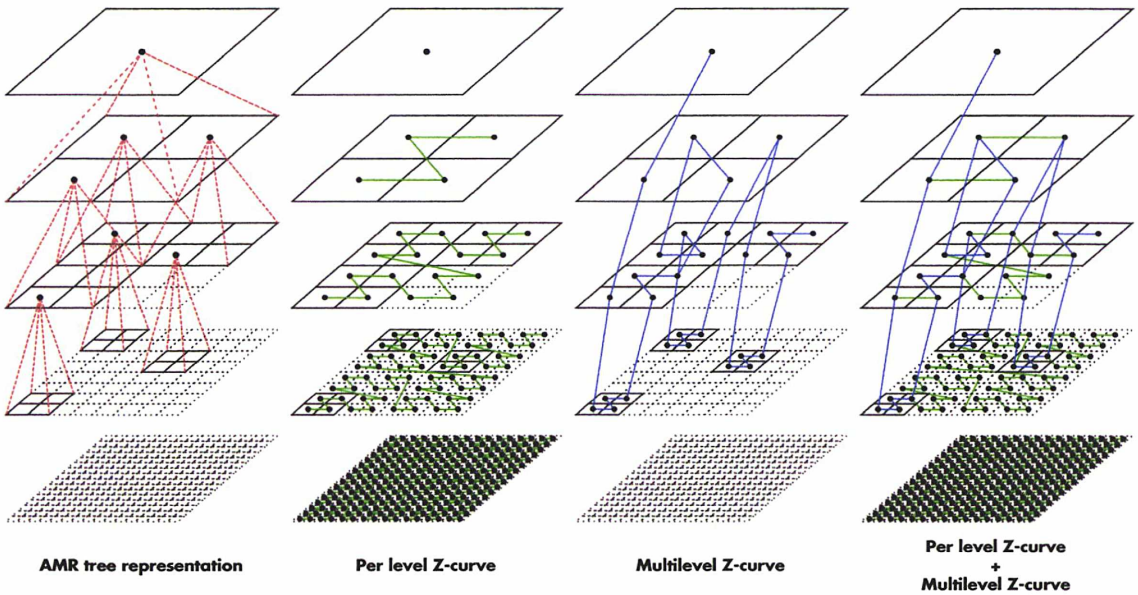


FIGURE 65 – Courbe de Lebesgue creuse multiniveaux. Á gauche, on considère un maillage adaptatif en dimension 2 à quatre niveaux, c'est-à-dire pour lesquels les cellules les plus raffinées sont au niveau 3. Pour encoder cette structure, se pose la question de l'indexation. La seconde figure montre comment les cellules peuvent être indexées par une courbe de Lebesgue, niveau par niveau. Cette méthode ne permet cependant pas d'obtenir une indexation globale : l'objectif est ici d'obtenir une courbe passant par toutes les cellules. Le troisième schéma représente une telle courbe, que l'on appelle dans la suite « courbe de Lebesgue creuse multiniveaux ». « Courbe de Lebesgue » car elle reprend le principe de la courbe verte comme l'indique la superposition de la quatrième figure. « Creuse », car à un niveau donné, elle ne traverse que les cellules qui existent : dans le cadre d'un raffinement adaptatif, elle n'est donc pas complète lorsque considérée niveau par niveau. Enfin « multiniveaux » car elle traverse toutes les cellules de tous les niveaux : pour un orthtree de dimension $N_{\text{dimensions}}$, elle est de dimension $N_{\text{dimensions}} + 1$. Comme indiqué en figure 66, une telle courbe émerge naturellement du tri du tableau d'indices obtenu à partir des mini-programmes encodant les positions des cellules.

étapes en avance, pour ne garder finalement que le branchement valide. Mais cette recherche présente un autre avantage : dans le cas d'un arbre non-creux, c'est-à-dire un arbre pour lequel chaque nœud est une feuille ou possède $2^{N_{\text{dimension}}}$ enfants, il est possible de trouver rapidement la grille la plus raffinée à n'importe quel endroit. Soit un point, ou une particule, dans l'espace réel à une position $\vec{x}$, il suffit de calculer l'indice i_{max} de la cellule la plus raffinée possible à cet emplacement en utilisant la méthode discutée à la sous-section précédente. La recherche binaire dans le tableau du plus grand indice vérifiant $i \leq i_{max}$ permet d'accéder directement au niveau le plus raffiné à cet endroit. La recherche des voisins, pour le calcul d'une interpolation Cloud-In-Cell par exemple, peut être effectuée avec la même approche.

Raffinement

Comme la structure en arbre émerge du tri des indices, le raffinement est plus coûteux puisqu'il faut insérer les cellules nouvellement créées à la bonne position. Du point de vue des performances d'insertion cela revient à comparer les performances d'un `std::vector` aux performances d'une `std::list`. L'insertion dans le premier cas scale en $\mathcal{O}(N)$ et en $\mathcal{O}(1)$ dans le second. Dans la pratique, pour des types fondamentaux comme des entiers, la copie est très rapide si bien que la différence de performances pour une insertion à une position aléatoire entre un tableau dynamique et une liste chaînée n'apparait que pour des très grands nombres d'éléments. Mais dans les faits, la question ne se pose de toute façon pas en ces termes, puisque les insertions sont généralement communes à un pas de temps. Autrement dit, il est possible de procéder à une phase de création de grilles pendant laquelle les nouveaux éléments sont ajoutés à la fin du tableau, puis à un unique tri commun pour rétablir la co-

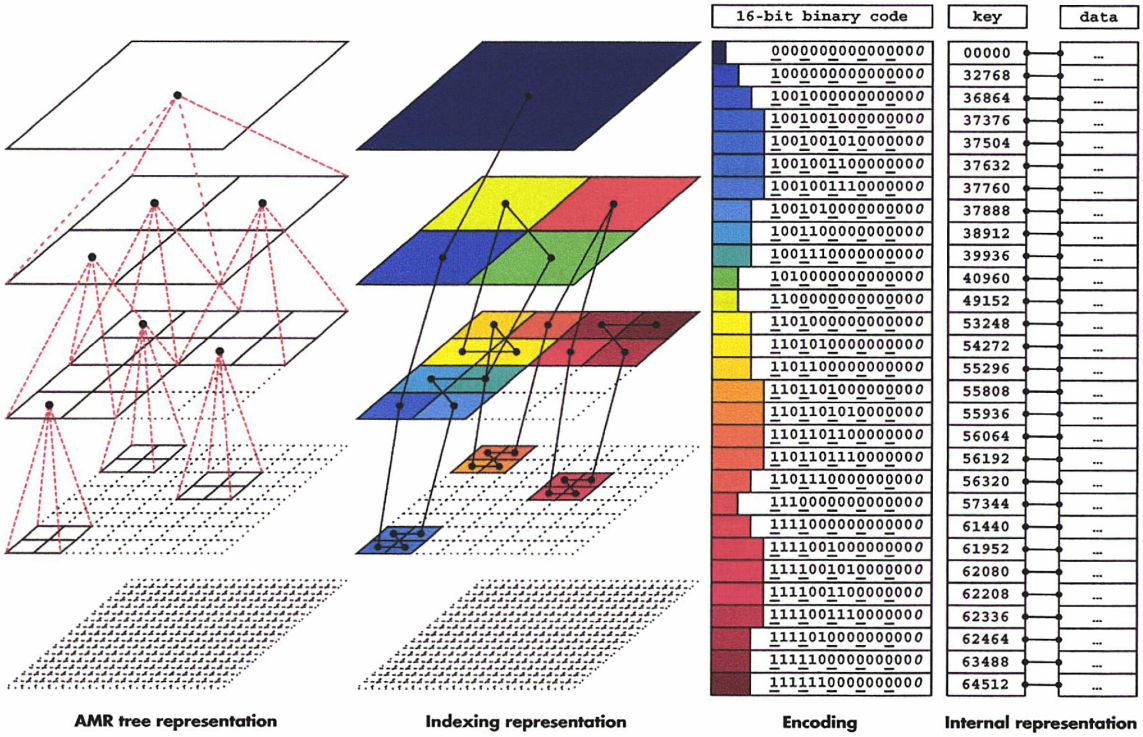


Figure 66 – Émergence des propriétés globales à partir du tri d'indices. La structure en arbre et la space filling curve émergent naturellement du tri du tableau d'indices. Cette émergence est illustrée ici dans le cas d'un maillage de raffinement adaptatif en deux dimensions. L'illustration se généralise aisément à un nombre arbitraire de dimensions et de niveaux. On considère l'arbre représenté dans la partie gauche de la figure. Pour chaque cellule, il est possible de calculer un indice conformément aux règles d'encodage de l'algorithme présenté dans ce chapitre. À l'issue de ce calcul, on range tous les indices obtenus dans un tableau, que l'on trie par ordre croissant selon la représentation entière des mini-programmes. On obtient ainsi le tableau de clés de la figure « internal representation ». La représentation binaire des indices est donné dans la colonne « 16-bit binary code » où les opcodes ont été soulignés, où les opérandes sont représentés normalement et où les bits non utilisés sont en italique. En associant une couleur à chaque indice du tableau, et en le reportant dans l'arbre, on voit la courbe de Lebesgue creuse multiniveaux de la figure 65 apparaître. La largeur des barres de couleur indique le nombre d'opcodes à 1, c'est-à-dire le niveau de raffinement. Il apparait alors que chaque cellule est directement suivie de ses enfants dans le cas où elle est raffinée, et de ses voisines sinon. Autrement dit, le tableau est déjà trié dans l'ordre d'un parcours en profondeur. Comme attendu, la structure en arbre émerge du tri du tableau : il suffit de regarder dans ce dernier pour savoir si une cellule est raffinée, pour accéder au parent de cette dernière, et pour accéder à ses voisins. Il s'agit d'un arbre implicite dans le sens où toute la structure est obtenue sans l'utilisation du moindre pointeur. Il ne reste alors plus qu'à stocker les données physiques associées à chaque indice de cellule dans un autre tableau trié de la même façon comme indiqué par la colonne « data ». Au final la structure obtenue est à la fois beaucoup plus légère en mémoire et beaucoup plus performante puisque les enfants sont proches de leur parent et puisque le tableau est trié selon un ordre classique de parcours d'arbre en profondeur.

hérence de l'arbre via un `std::inplace_merge`. De la même manière, la suppression de cellules peut être effectuée en égalisant les indices des nœuds à supprimer à ceux de leur parent puis en procédant à un unique `std::unique` pour supprimer les indices redondants et rétablir la cohérence de l'ensemble.

Conclusion

Comme nous venons de le voir, le tri des indices préalablement obtenus permet de faire émerger un arbre implicite exhibant des bonnes propriétés du point de vue de l'empreinte mémoire et des performances. Le problème de la fragmentation ne se pose pas et la structure de données est compatible avec le cache et le prefetcher. Au parcours de pointeurs en pointeurs se substitue une recherche binaire en mémoire sur les indices. La création et la destruction de cellules pourraient poser problème en termes

de performances si elles intervenaient à des instants arbitraires, mais elles arrivent généralement par phase, ce qui permet une insertion globale optimisée à base de `std::inplace_merge` plutôt que de multiples insertions répétées.

6.2.4 Avantages et limitations

Introduction

L'algorithme présenté dans ce chapitre permet d'aborder le problème des arbres de maillage adaptatif d'une nouvelle façon. Ce faisant, on règle un certain nombre de problèmes des implémentations classiques : fragmentation de la mémoire, limitation des optimisations possibles du fait des indirections, empreinte mémoire importante, complexité de la parallélisation au moyen de space filling curves...

Empreinte mémoire

Tout d'abord du point de vue de l'empreinte mémoire, la réduction provient du fait que d'une part, la position de la cellule est encodée dans un entier et que, d'autre part, les liens entre cellules n'ont pas besoin d'être matérialisés sous forme de pointeurs, l'ordre dans lequel elles sont triées dans le tableau étant suffisant. On économise donc la mémoire sur ces deux plans simultanément. Dans un contexte où la mémoire est un facteur limitant des simulations cosmologiques, il s'agit d'un point nodal pour l'avenir des simulations sur les calculateurs exascale dans le sens où la mémoire disponible par cœur de calcul semble stagner voire régresser. Au final, l'algorithme dont il est question ici réduit l'empreinte de l'arbre à son minimum : un entier par cellule. Il faut toutefois remarquer deux choses sur l'encodage. Premièrement, la compression de l'information n'est pas maximale dans le sens où certains indices n'ont pas de sens : $1101100100000000 = 55552_{10}$ peut ainsi exister alors que $1101100101100000 = 55648_{10}$ n'est pas valide car des bits non nuls suivent le premier opcode nul. Cette non utilisation de certaines plages d'indices peut être vue comme une non optimalité de l'algorithme. Deuxièmement, c'est le nombre de bits des entiers N_{bits} qui conditionne le niveau maximal de raffinement à nombre de dimension $N_{\text{dimensions}}$ fixé :

$$N_{\text{level}} = \frac{N_{\text{bits}}}{N_{\text{dimensions}} + 1} \tag{6.5}$$

Par exemple, pour des entiers 64 bits et trois dimensions d'espace, on peut atteindre au maximum 16 niveaux de subdivision depuis le niveau 0. Aussi, pour les analyses du Full Universe Run, qui totalise 19 niveaux de raffinements [9], des entiers 128 bits non standard `__uint128_t` ont été choisis. De tels entiers peuvent permettre d'atteindre jusqu'à 32 niveaux, ce qui est actuellement suffisant pour toutes les simulations cosmologiques standard en dimension 3 puisque cela correspondrait à une résolution spatiale d'environ 5 pc/h, soit l'ordre de grandeur de la distance inter-étoiles dans la Galaxie, pour une simulation de la taille de l'Univers Observable de 21 Gpc/h de côté. Le point important de cette remarque est que l'algorithme d'encodage de l'information ne fonctionne que parce que l'arité [10] et la profondeur maximal de l'arbre sont connus et fixés à la compilation. C'est cela qui permet à l'algorithme de profiter d'un encodage efficace et qui le rend parfaitement adapté aux simulations cosmologiques. Mais cela limite aussi ses domaines d'application.

Maillage de l'espace

On l'a vu, au niveau de la structure de l'arbre, il est nécessaire de connaître à l'avance le nombre maximal d'enfants, ce qui est le cas des orthtrees, et le niveau maximal. Il existe deux autres limitations. D'une part, l'encodage nécessite de systématiquement débuter du niveau 0 : ainsi on ne peut tirer parti de la présence d'un niveau « coarse » pour compresser l'information à moins bien sûr de significativement modifier l'encodage. D'autre part, la structure de l'arbre implicite est selon une courbe de

9. En fait 6 niveaux depuis le niveau coarse $8192 = 2^{13}$.
10. Nombre maximal d'enfants par nœud.

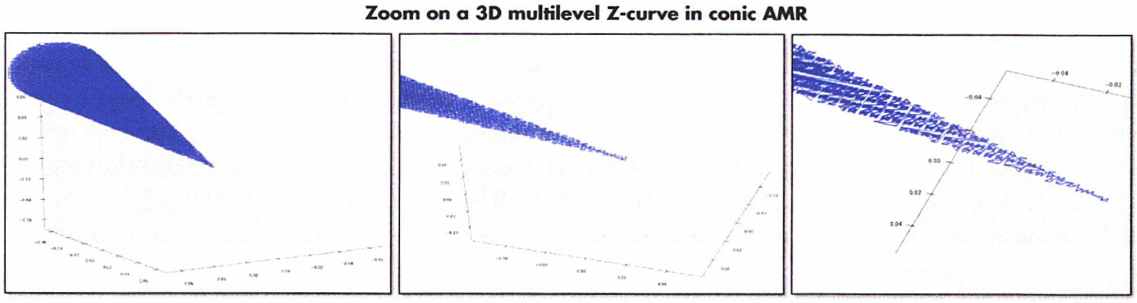


FIGURE 67 – Courbe de Lebesgue multiniveaux dans un cône à base circulaire. L'algorithme ne nécessite pas que l'arbre soit complet pour fonctionner. Dès lors, il est possible de construire des maillages hypercubiques pour des formes arbitraires pour lesquels le nombre de cellules filles est inférieur à $2^{N_{\text{dimensions}}}$, pour lesquels il existe un niveau grossier différent du niveau 0 et pour lesquels tout l'espace n'est pas maillé. Ici, on représente à titre d'exemple la courbe de Lebesgue multiniveaux pour les cellules composant un cône à base circulaire utilisé pour les analyses décrites dans la partie physique de ce manuscrit.

Lebesgue, qui est moins optimale que la courbe de Hilbert du point de vue de la localité des données. Si la première est structurellement au cœur de l'algorithme et présente les nombreux avantages énoncés dans les sous-sections précédentes, il est nécessaire de garder en tête cette non optimalité lorsqu'il est question de parallélisation et d'optimisation des communications sur les supercalculateurs. Mais la structure de l'arbre présente d'autres avantages puisqu'elle ne se limite pas à des orthtrees complets. En effet, si l'encodage des indices nécessite d'enregistrer la navigation depuis le niveau 0, la présence de ces niveaux dans le tableau n'est pas nécessaire : l'algorithme fonctionne parfaitement avec un niveau « coarse » sans cellules parents. D'autre part, l'arbre peut être creux, dans le sens où il n'est pas nécessaire que chaque cellule non feuille soit divisée en exactement $2^{N_{\text{dimensions}}}$ enfants : tous les enfants n'ont pas à être présents pour que la structure de l'arbre soit cohérente. Pour certaines analyses, cela peut avoir son intérêt pour économiser de la mémoire. Il n'est en fait pas non plus nécessaire de paver tout l'espace. Et cela représente un avantage significatif puisqu'il devient possible de mailler n'importe quelle forme avec des hypercubes. Dans le cadre de ce manuscrit, on peut par exemple penser à des cônes : comme il n'est ni nécessaire de commencer du niveau 0, ni que l'arbre soit complet, ni que tout l'espace soit pavé, on peut ainsi obtenir une courbe de Lebesgue multiniveaux pour des arbres de raffinement adaptatif sur des cônes à base circulaire comme illustré en figure 67.

Encodage des indices

Concernant l'encodage lui-même, ce dernier présente plusieurs avantages. Tout d'abord, le gain en mémoire énoncé plus tôt n'a qu'un impact très limité sur les performances. En effet, comme nous l'avons vu, les nouveaux jeux d'instructions de manipulations de bits tels que BMI permettent de limiter l'encodage et le décodage des indices à quelques instructions assembleur. L'algorithme est donc, de fait, parfaitement optimisé pour les futures architectures qui proposeront ces jeux d'instructions. Nous avons montré comment il était ainsi possible de calculer simplement les indices des parents, des enfants, et des voisins, comment il était possible d'extraire le niveau de raffinement et les coordonnées, et comment il était possible d'effectuer les conversions vers et à partir des indices monodimensionnels et des indices des courbes de Lebesgue à niveau fixé. Contrairement au cas de RAMSES où les coordonnées des cellules sont stockées sous forme flottante avec une précision qui dépend de la position, les coordonnées sont stockées ici sous forme entière ce qui permet d'éviter naturellement ces problèmes de précision. De plus, l'utilisation de nombres entiers non signés pour les indices permet de tirer parti du complément à deux et de l'arithmétique en modulo de 2^N : lorsque les coordonnées dépassent le nombre de cellules du niveau, ces coordonnées sont traitées naturellement comme s'il y avait une périodicité de la boîte. Autrement dit, si l'on cherche la cellule à l'emplacement $(1.2, 0.5, 4.3)$ en coordonnées réelles normalisées à 1, l'injection de ces nombres réels dans la procédure de calcul

des indices retournera naturellement la grille à la position $(0.2, 0.5, 0.3)$ sans qu'il ait été nécessaire d'appliquer la moindre correction : l'encodage périodise ainsi implicitement l'espace. Dans le cadre de simulations cosmologiques avec des conditions aux bords périodiques cette propriété peut s'avérer être très appréciable. Enfin, le fait d'associer à chaque ensemble de données physiques une clé unique à partir de laquelle il est possible de reconstituer la position dans l'espace a un intérêt du point de vue du stockage et de la gestion des données. En effet, dès lors que l'on utilise cette méthode d'enregistrement, il suffit de connaître une clé pour connaître la situation spatiale d'une donnée sans avoir à stocker les détails d'une structure en arbre complexe. Le seul problème avec cette stratégie est que les mini-programmes sont stockés sous forme entière en commençant par les bits de poids forts : un même programme avec des choix de N_{bits} différents sera interprété comme deux valeurs différentes. Une solution pour contourner ce problème, consiste à procéder à une opération de permutation des bits avant chaque sauvegarde pour faire en sorte que l'information soit stockée en commençant par les bits de poids faible.

Cache et prefetching

Les gains en performances proviennent également du fait qu'une structure à base d'indirections et de pointeurs est ici remplacée par un simple tableau au sein duquel les données sont triées dans un ordre de parcours en profondeur préfixé. Lorsqu'il s'agit de passer en revue l'ensemble des données, l'on passe ainsi d'une navigation complexe au simple parcours linéaire d'un tableau. Rien que de ce point de vue, le gain est significatif. D'autre part, ce tableau respecte une certaines localité des données : des données proches spatialement sont proches dans la mémoire, du moins beaucoup plus que dans l'approche classique. En conséquence, la structure de données offre une bien meilleure efficacité au niveau du cache, puisque pour travailler sur des portions d'espace il suffit de transférer des portions contigües de mémoire en cache. Dans un contexte où le transfert de la mémoire vive au processeur est appelé à devenir de plus en plus coûteux comparativement au temps de calcul, il s'agit là d'un avantage non négligeable. Le prefetcher peut également bénéficier de cela puisque l'ordre de parcours de données peut devenir prévisible, qu'il s'agisse de lire le tableau linéairement ou qu'il s'agisse d'y effectuer une recherche binaire pour laquelle les branchements en mémoire peuvent être pré-évalués plusieurs étapes en avance. En effet, dans le cadre de cet algorithme, une recherche de cellule est remplacée d'une navigation non prédictible de pointeurs en pointeurs à une recherche binaire sur des nombres entiers triés par ordre croissant. Des premiers benchmarks qu'il conviendrait de compléter par des études plus approfondies semblent ainsi indiquer des gains entre 5x et 20x sur le parcours et la recherche de données dans le tableau : on est dans tous les cas bien loin d'optimisations impliquant des gains de l'ordre de quelques pourcents. On gagne donc à la fois en termes de mémoire et en termes de performances. L'une des limitations vient du fait que la création et la destruction de cellules devient plus coûteuse. Cela devient effectivement le cas lorsque les indices sont stockés dans une zone de mémoire contigüe : l'insertion de nouveaux éléments évolue alors en $\mathcal{O}(N)$. L'impact du problème reste toutefois limité par le fait qu'il s'agit de travailler sur des indices entiers dont la recopie en mémoire est extrêmement rapide, et surtout que l'insertion et la suppression d'éléments de grille interviennent par phase de raffinement. Ainsi, la cohérence du tableau peut n'être rétablie qu'une seule fois par pas de temps dans le cadre d'une simulation dynamique.

Parallélisation et vectorisation

Comme nous l'avons vu précédemment, une propriété intéressante de l'arbre implicite est de disposer naturellement d'une space filling curve : en l'occurrence d'une courbe de Lebesgue multiniveaux. Cela ouvre les portes de gains élevés en termes d'efficacité et de simplicité de parallélisation. Sur la simplicité tout d'abord : on passe d'une situation où il est nécessaire de calculer une courbe de Hilbert sur des portions d'espace complexes et de communiquer une structure fractionnée en mémoire, à une situation où il suffit de recopier une portion de tableau d'un processus à un autre. En effet, en reprenant le schéma de la figure 66, il apparait que distribuer le travail sur plusieurs processus revient simplement

à distribuer le contenu du tableau sur plusieurs tâches. Bien entendu, il ne s'agit pas de découper ce dernier à n'importe quel endroit pour ne pas séparer une cellule parent de ses cellules enfants : mais en s'assurant de maintenir cette cohérence et en s'assurant de recopier les niveaux plus grossiers que le niveau utilisé pour la « découpe », la parallélisation sur des machines à mémoire distribuée devient bien plus simple qu'avec un orthtree classique. D'autre part, le portage de codes de raffinement adaptatif sur des accélérateurs tels que des cartes graphiques ou des unités manycores reposant massivement sur la vectorisation est aujourd'hui largement problématique. Ainsi, ce sont généralement des versions des codes sur grille fixe qui sont portés sur des architectures hétérogènes de cette nature. Ici, il n'y a pas de fragmentation mémoire, et la complexité de la structure de données reste la même qu'il s'agisse d'une grille à un seul niveau ou d'une grille raffinée. Du point de vue d'unités graphiques ou de cartes largement basées sur les gains potentiels apportés par le SIMD [11], l'avantage semble significatif. Si cette voie n'a pour le moment pas été explorée, il apparait cependant comme fortement envisageable que l'algorithme décrit dans ce chapitre permette de gérer relativement simplement des orthtrees raffinés sur des unités de vectorisation.

Accès aux données lagrangiennes

Un dernier avantage concerne la gestion des données lagrangiennes, c'est-à-dire les particules dans le contexte de la cosmologie numérique. Tout ce qui a été décrit précédemment ne concerne en principe que les quantités physiques eulériennes, distribuées sur la grille. Or il apparait que l'algorithme présente aussi une particularité appréciable pour la gestion des particules qui naviguent de cellule en cellule. D'un point de vue de l'architecture logicielle la question de la gestion des particules au sein de la grille est importante : une première possibilité consiste à stocker les particules dans les cellules auxquelles elles appartiennent au moyen d'un simple tableau dynamique pour chaque cellule, une seconde consiste à lister toutes les particules indépendamment et à sauvegarder l'information de la cellule à laquelle elles appartiennent. Mais dans ce dernier cas, comme pour les quantités eulériennes, cela pose un problème d'indirection puisque les particules ne sont stockées en mémoire dans aucun ordre particulier. L'approche mini-programme/nombre entier permet d'apporter une solution à ce problème. La position de chaque quantité lagrangienne est donnée par $N_{\text{dimensions}}$ réels. Or, à partir de cette position, il est aisé de calculer en quelques instructions assembleur l'indice de la cellule la plus raffinée possible à cet endroit comme exposé dans la sous-section précédente. Si l'on stocke les particules dans un unique tableau, comme le sont les cellules, et que l'on trie ce tableau en fonction de cet indice, alors les particules apparaissent en mémoire dans le même ordre que les cellules de l'arbre. Ce faisant, on fait passer une courbe de Lebesgue par les particules au niveau le plus raffiné possible. Il suffit alors de parcourir simultanément le tableau de cellules et le tableau de particules pour savoir quelles particules appartiennent à quelles cellules : les deux types d'entité apparaissent dans le même ordre.

Conclusion

Ainsi, toutes les limitations classiques existantes évoquées en introduction de ce chapitre semblent trouver une réponse dans la dualité mini-programme/nombre entier et l'arbre implicite qui en résulte : empreinte mémoire, performances, efficacité du cache et parallélisation avec des gains qui dépassent à chaque fois très largement l'ordre de la dizaine de pourcents. Cela ouvre donc des perspectives d'améliorations significatives du niveau « statique » qui est au cœur des codes de simulation cosmologique. Bien entendu, la solution parfaite n'existe pas et plusieurs limitations de la nouvelle méthode ont ici été évoquées. Elle permet cependant de gagner sur beaucoup de tableaux simultanément et est d'autant plus intéressante qu'elle semble parfaitement adaptée à des architectures disposant de jeux d'instructions sur les manipulations de bits et à des architectures hétérogènes disposant de cartes tirant parti de la vectorisation. Deux caractéristiques qui selon toute vraisemblance seront au rendez-vous dans le contexte de l'exascale.

11. Le SIMD, pour « Single Instruction Multiple Data », désigne ici les capacités de traitement vectoriel des processeurs.

6.3 Application et perspectives

6.3.1 Implémentation et améliorations

Une version parfaitement fonctionnelle de l'algorithme a été développée pour la gestion de l'arbre du Full Universe Run dans le cadre de MAGRATHEA pour les analyses de propagation de la lumière. Les opérations d'encodage et de décodage des indices ont pour le moment été implémentées à grand renfort de métaprogrammation sans faire appel aux instructions natives de manipulation de bits, ces dernières n'étant pas encore suffisamment répandues. Il s'agit donc d'une version non dépendante des jeux d'instructions des architectures. Cela signifie également que des gains substantiels en termes de performances sont encore à prévoir. Cette implémentation, si elle n'a pas été développée dans un esprit de version finalisée, a toutefois permis de s'assurer de la viabilité pratique de l'algorithme et de s'assurer que les gains en mémoire et en performances étaient bien au rendez-vous. Une étude détaillée des performances dans le cadre de benchmarks comparatifs de différentes implémentations d'orthtrees est aujourd'hui à envisager. Du point de vue de la viabilité, la gestion de centaines de milliards de cellules sur plusieurs milliers de cœurs n'a révélé aucun problème particulier de scalabilité. À l'heure actuelle, une parallélisation hybride de l'arbre MPI/`std::thread` a été implémentée : MPI pour la distribution sur les nœuds de calcul, et `std::thread` pour la parallélisation sur les unités à mémoire partagée. Là encore, la mise en pratique des éléments de la partie précédente n'a révélé aucun problème de fond.

L'implémentation d'une version exploratoire a également permis d'envisager des manières de repousser certaines des limitations de l'algorithme. L'un des problèmes au niveau des performances est lié à la scalabilité en $\mathcal{O}(N)$ de l'insertion pour de grands tableaux. Dans une approche hybride avec une portion commune d'arbre par nœud de calcul, la taille de l'arbre local peut être significative et cela peut commencer à poser des problèmes de performances. À cela s'ajoute le problème de l'encodage depuis le niveau 0 et l'impossibilité de compresser l'information lorsque la simulation dispose d'un niveau « coarse », ce qui est très généralement le cas. Ces deux limitations peuvent être dépassées par une unique approche : voir une simulation comme une forêt d'orthtrees, ou comme un orthtree d'orthtrees et non comme un seul arbre monolithique. Ainsi le Full Universe Run peut être vu comme une grille fixe de 8192^3 arbres de raffinement adaptatif indépendants. Ce faisant, l'insertion ou la recherche de cellules ne pose plus aucun problème de performances puisque le nombre de cellules par arbre est drastiquement réduit. Cela permet aussi de gérer l'accès aux voisins de manière optimisée car l'emplacement mémoire des données au sein d'une grille fixe est parfaitement connu dès l'initialisation. Les performances sont donc encore accrues comparativement à l'utilisation d'un arbre unique. On gagne également sur l'occupation mémoire puisque le niveau 0 de chaque arbre correspond au niveau grossier de la simulation. Ainsi, avec 6 niveaux de raffinement au-delà du niveau coarse, la grille du Full Universe Run peut être encodée au moyen d'entiers 32-bit et non d'entiers 128-bit comme actuellement. Dans ce cas, on améliore donc encore l'empreinte mémoire d'un facteur 4. De manière plus générale, procéder de la sorte permet aussi de paver des espaces potentiellement bien plus complexes que des cubes, puisqu'il ne s'agit plus de mailler des simulations avec un seul orthtree, mais de les mailler avec une multitude d'orthtrees.

Pour terminer, cette méthode donne aussi accès à une parallélisation plus performante en rendant possible l'utilisation d'une courbe de Hilbert. En effet, à partir du moment où l'on dispose d'une forêt d'orthtrees, rien n'empêche de calculer la courbe de Hilbert pour cette forêt et de baser les schémas de communication sur cette courbe. Cela annule la légère perte de performances qui provenait de la non-optimalité de la localité des données fournie par la courbe de Lebesgue. Dans un tel contexte, lors d'un rééquilibrage de charge, il n'est plus question de communiquer des cellules d'une tâche à l'autre, mais de communiquer des orthtrees complets. Et cela reste excessivement simple, puisqu'il s'agit de copier un tableau d'indices et un tableau de données d'un processus à l'autre.

Ainsi, l'implémentation actuelle a d'ores et déjà permis de s'assurer de la viabilité de l'algorithme,

y compris dans un cadre applicatif lourd à savoir l'analyse de la propagation de la lumière au sein du Full Universe Run. Des tests de performances et de scalabilité détaillés restent à faire. Au-delà de la viabilité, cela a conduit à poser les principes d'une structure encore plus performante à base de forêts d'orthtrees. Autrement dit une structure hybride avec une parallélisation hybride pour s'adapter au mieux à l'architecture des machines actuelles et à venir. Cette voie reste à l'heure actuelle théorique et devrait faire l'objet d'explorations algorithmiques futures.

6.3.2 Vers une généralisation à tout type d'arbre

La réflexion menée ici sur les orthtrees invite également à s'interroger sur un problème plus global : celui des arbres. Il s'agit là d'une structure de données essentielle dans de nombreux de domaines applicatifs, et notamment dans beaucoup de champs astrophysiques basés sur l'exploitation de données distribuées dans l'espace. Serait-il possible de généraliser l'algorithme présenté dans les paragraphes précédents à tout type d'arbre ?

Poser cette question, c'est aussi poser celle de l'unification des arbres. Si l'on s'intéresse à l'évolution de la bibliothèque standard du C++ ou à l'évolution de Boost, il apparait qu'une bibliothèque d'arbres n'a jamais vu le jour pour la simple raison qu'il existe des dizaines d'arbres différents et qu'aucune structure ne peut répondre à tous les besoins. Autrement dit, l'unification de la variété existante d'arbres dans le paysage des structures de données est aujourd'hui un problème ouvert. Une solution possible consiste à se concentrer sur la généralisation des arbres d'un point de vue mathématique à savoir les graphes. C'est l'approche qui a conduit à l'élaboration de la BGL, la Boost Graph Library[137]. On se place alors dans un contexte très général, dans lequel les arbres ne sont qu'un cas particulier de graphes : le cas de graphes non orientés, acycliques et connexes. L'avantage de cette formulation du problème est qu'elle permet effectivement de spécifier une structure suffisamment générique pour répondre aux besoins de tous les utilisateurs. Les deux inconvénients majeurs sont que d'une part, le vocabulaire spécifique des arbres disparait au profit du vocabulaire des graphes, ce qui est déroutant pour un certain nombre de domaines applicatifs, et que d'autre part, de nombreuses optimisations liées au fait de disposer de graphes non orientés, acycliques et connexes sont ignorées. Dès lors, même avec l'existence de la BGL, la question d'une bibliothèque d'arbres génériques reste digne d'un très grand intérêt.

Pour résoudre le problème de l'unification, il est nécessaire de mettre en lumière des régularités dans la variété d'arbres existants. La consultation de liste d'arbres usuels peut mettre en lumière plus d'une centaine de type d'arbres différents. Il est important de préciser que l'on ne se pose pas ici la question d'une unification théorique, mais d'une unification pratique : si une bibliothèque permet de satisfaire les besoins de 90% des applications basés sur les arbres, alors l'objectif d'unification peut être considéré comme atteint, les cas à la marge pouvant de toute façon être réexprimés dans le cadre de la théorie des graphes et implémentés avec la BGL. Les paragraphes suivants donnent quelques pistes de réflexion basées sur le travail exposé dans ce chapitre. On ne considère dans la suite que le cas de « rooted trees », c'est-à-dire les arbres dont un nœud a été identifié comme la racine.

Tout d'abord, en termes d'implémentation, qui conditionne les performances, il est possible de différencier deux grandes catégories :

- les arbres explicites
- les arbres implicites

Les premiers sont ceux basés sur des pointeurs et sur une représentation « classique » des liens entre les nœuds, tandis que dans le second cas, la structure en arbre émerge naturellement d'une structure plus fondamentale, comme ce qui a été exposé ici. Il est ensuite possible de distinguer quatre limitations qui conditionnent un grand nombre d'optimisations :

- les arbres à arité et à profondeur maximale statiques

- les arbres à arité statique et à profondeur dynamique
- les arbres à arité dynamique et à profondeur statique
- les arbres à arité et à profondeur maximale dynamiques

où statique est synonyme de « fixée à la compilation » et où dynamique est synonyme de « libre à la compilation ». En combinant ces quatre catégories aux deux précédentes, on obtient ainsi huit types de structures. Et force est de constater que ces distinctions permettent d'implémenter de façon optimale un très grand nombre d'arbres. Dans le cas de l'algorithme décrit précédemment on traite d'arbres implicites à arité et profondeur maximale statiques. Les kd-trees[138] utilisés par exemple pour le calcul de fonctions de corrélations dans des grands relevés observationnels s'inscrivent également dans cette catégorie. Si l'on relâche la limite sur la profondeur maximale, alors on peut tout à fait imaginer une implémentation avec le même encodage, mais avec des entiers de taille variable. Si l'on relâche la limite sur l'arité, alors il devient nécessaire de stocker la navigation à chaque niveau en clair. Dans ce cas, la représentation explicite est d'ailleurs surement plus adaptée.

En se basant sur ces distinctions, il devient envisageable de créer un arbre générique `basic_tree` dont les paramètres template viendront contraindre les propriétés énoncées précédemment. Ce faisant, l'on disposerait de structures optimisées pour la plupart des cas rencontrés, l'élaboration de structures plus évoluées reposant sur la spécialisation, l'héritage et la composition de cette structure de données fondamentale. Ces pistes de réflexion nécessiteraient bien entendu une étude plus aboutie, mais dans un contexte où les arbres sont omniprésents pour le traitement des données astrophysiques, il apparaîtrait comme profitable pour toute la communauté de réfléchir à une structure de données unifiée répondant de manière optimale à la plupart des besoins.

Chapitre 7

Vers un code de raytracing en métrique faiblement perturbée

***

7.1 Élements de conception 156
7.1.1 Algorithme 156
7.1.2 Architecture et implémentation 160
7.1.3 Parallélisation 161
7.2 Tests et vérifications 164
7.2.1 Construction de l'arbre 164
7.2.2 Intégration dans le cas homogène 165
7.2.3 Intégration en métrique de Schwarzschild 166
7.3 Application cosmologique 172
7.3.1 Intégration des géodésiques dans les cônes de lumière du Full Universe Run . . 172
7.3.2 Perspectives 173

Les chapitres précédents nous ont invité à nous réinterroger sur la conception des codes cosmologiques suite aux leçons tirées de la réalisation du Full Universe Run. Pour tenter d'apporter une réponse aux problèmes soulevés nous avons exploré la piste de la réalisation de bibliothèques actives basées sur la métaprogrammation. Nous avons aussi vu comment la structure des codes de simulation pouvait être réexprimée en trois niveaux de complexité : statique, cinématique et dynamique. Le chapitre précédent se concentrait sur la conception d'un nouvel algorithme de gestion du raffinement adaptatif, ce dernier étant un élément central de la composante statique. Nous nous penchons maintenant sur la partie cinématique dans le cadre d'un code visant à intégrer numériquement les équations géodésiques en métrique faiblement perturbée pour traiter la propagation de la lumière au sein du Full Universe Run. L'implémentation de ce code s'appuie à la fois sur la bibliothèque Magrathea et sur le nouvel algorithme de maillage adaptatif. En plus d'être à même de produire les résultats physiques présentés au quatrième chapitre, ce code peut être vu comme l'application à un cas pratique des méthodes et techniques présentées dans les chapitres précédents.

Ce chapitre, qui se veut donc comme un complément numérique du chapitre quatre, se divise en trois sous-parties. La première résume les principaux aspects de conception du code : y sont décrit l'algorithme principal et les différents éléments constitutifs de ce dernier. La seconde s'attache à vérifier le bon fonctionnement de l'ensemble : pour cela un certain nombre de tests sont effectués dans des configurations où la solution physique est connue. Cela permet au passage d'évaluer les erreurs numériques de l'algorithme. Un accent particulier est mis sur la solution en métrique de Schwarzschild qui

constitue un cas de référence particulièrement pertinent. Un développement de la solution analytique directement comparable à la solution numérique est proposé à cette occasion. Enfin, la dernière partie revient quelques éléments important de la mise en application du code dans le cadre de l'analyse des cônes de lumière du Full Universe Run.

7.1 Élements de conception

7.1.1 Algorithme

Introduction

Le but est ici d'intégrer numériquement les géodésiques au sein de simulations cosmologiques en vue notamment d'analyser les cônes de lumière du Full Universe Run. Le principe de cette intégration a été présentée au chapitre quatre : il s'agit de calculer les trajectoires de photons se propageant depuis un observateur central jusqu'à des redshifts de plus en plus grand, et ce, sur toute la sphère céleste. Ces photons sont regroupés en faisceaux afin d'être en mesure de calculer les distances cosmologiques. Le code développé spécialement pour l'occasion en C++11 et décrit ici s'appuie à la fois sur la bibliothèque active MAGRATHEA et sur une implémentation de l'algorithme de maillage adaptatif décrit au chapitre précédent.

Vue globale

Un résumé global de l'algorithme du code d'intégration des géodésiques en champ faible est présenté en figure 68. Ce dernier est basé sur une parallélisation hybride à deux niveaux : MPI pour le parallélisme inter-nœuds et les threads de la bibliothèque standard du C++11 pour le parallélisme inter-nœuds. Le premier consiste en un découpage spatial de la simulation, tandis que le second consiste en une répartition des groupes de photons sur les cœurs disponibles localement. La figure 68 résume le processus global pour chaque tâche MPI. L'algorithme commence par une phase d'initialisation des paramètres d'exécution en provenance d'un fichier de paramètres et en provenance de la table d'évolution cosmologique correspondant à la simulation. Une seconde phase consiste en l'acquisition des données physiques : c'est-à-dire de la distribution de potentiel dans l'espace. Les fichiers de données bruts RAMSES sont lus, chargés en mémoire, et utilisés comme base pour reconstruire un arbre hypercubique de la portion d'espace traitée par chaque tâche. Ensuite, quatre choix sont possibles suivant le mode sélectionné dans le fichier de paramètres : la configuration d'exécution cosmologique et trois types de tests pour la vérification de l'algorithme : le premier pour vérifier la structure et la construction de l'arbre, le second pour vérifier l'intégration dans un cadre homogène et le dernier, enfin, pour tester l'intégration dans le cadre d'une métrique de Schwarzschild. La configuration d'exécution normale permet de boucler sur plusieurs paramètres numériques : le nombre de photons par « bundle » [1], l'angle d'ouverture de chaque « bundle », les conditions d'arrêt décrites au quatrième chapitre et les quantités statistiques à calculer. Grâce à cela, il est possible d'explorer en une seule exécution plusieurs paramètres numériques importants. Pour chacune des itérations sur cet ensemble de paramètres, les géodésiques sont intégrées en utilisant un intégrateur de Runge-Kutta d'ordre 4 pour une grande quantité de groupes de photons. La déformation de chaque « bundle » permet ensuite de calculer l'évolution de la distance de diamètre angulaire. À l'issue de ces calculs, le programme peut soit sauvegarder les informations sur chaque bundle indépendamment, soit calculer des grandeurs statistiques sur ces derniers.

Initialisation et acquisition des paramètres

Au démarrage du programme, l'ensemble des structures de données sont initialisées. L'arbre à construire destiné à accueillir les données physiques pré-alloue une zone de mémoire conséquente

1. Faisceau.

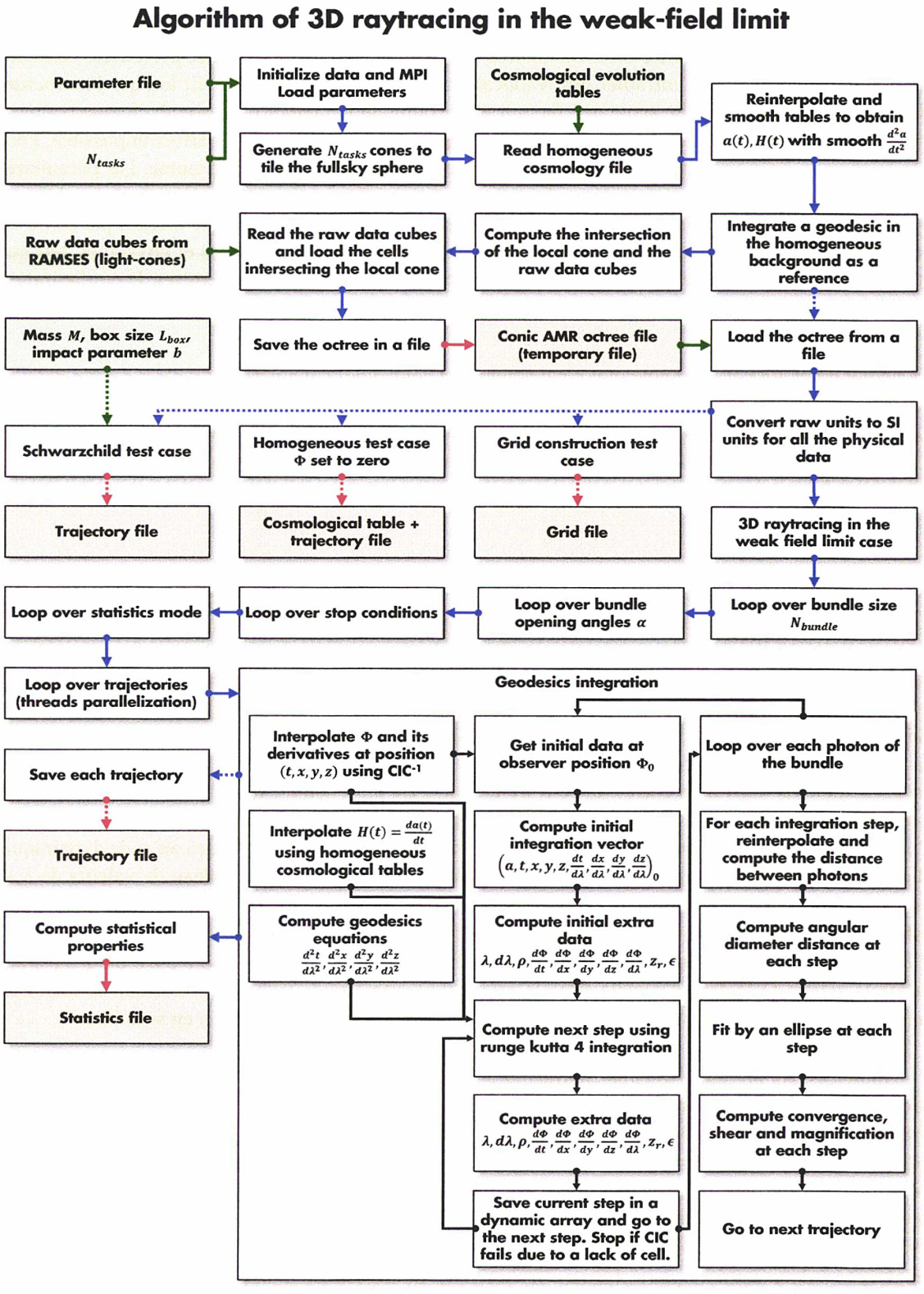


FIGURE 68 – **Algorithme sous-tendant le code d'intégration des géodésiques en régime de champ faible.** Les éléments en vert indiquent les paramètres et les fichiers d'entrée du code, les éléments en rouge constituent les fichiers de sortie et les pointillés indiquent les branches optionnelles, hors de l'intégration standard. L'algorithme fait l'objet d'une parallélisation hybride, où chaque tâche hérite d'un domaine conique de la sphère céleste originale et où la boucle sur les groupes de trajectoires est parallélisée via du threading. Pour faciliter l'intégration avec de multiples paramètres et le redémarrage du code, l'arbre AMR conique est enregistré dans des fichiers temporaires lors de la première passe.

dont la taille est paramétrable. En effet, comme ce dernier se présente sous la forme de tableaux dynamiques contigus en mémoire conformément à ce qui a été présenté au chapitre précédent, l'absence de pré-allocation pourrait entraîner des réallocations au cours de l'exécution. Et lorsque la structure de données représente à elle seule une faction importante de la mémoire du nœud, chaque réallocation est potentiellement coûteuse en temps de recopie. Elle peut même être parfois impossible. Pour ces raisons, la mémoire de l'arbre est donc réservée à l'initialisation du programme. Les paramètres concernant la parallélisation, eux, dépendent des conditions d'exécution : le nombre de tâches MPI est configuré lors du lancement des travaux sur la machine, tandis que le nombre de threads par tâche est récupéré de façon automatique. L'ensemble des autres paramètres, numériques et physiques, tels que les chemins d'accès vers les fichiers, le mode d'exécution, le nombre de trajectoires, la configuration des bundles... sont spécifiés dans un fichier texte indépendant sous la forme :

```
1 key = value
```

Le fichier est chargé en mémoire, puis chaque ligne est contrôlée et analysée de manière à extraire la clé et la valeur. Ces dernières sont ensuite injectées dans une `std::map<std::string, std::string>`. La conversion des chaînes de caractères vers des valeurs numériques est effectuée au cas par cas. À l'issue de cette procédure l'ensemble des variables associées à la configuration de la simulation sont donc initialisées.

Acquisition de la cosmologie homogène

Trois quantités liées à la cosmologie homogène et non spécifiées dans les fichiers bruts de données issus de la simulation sont nécessaires pour effectuer les bonnes conversions d'unité : la taille de boîte L_{box} en Mpc/h, la constante de Hubble h et le paramètre de densité pour la matière Ω_{M_0}. Le premier est récupéré automatique par le programme par une analyse des noms de fichiers, qui comporte généralement cette taille de boîte. Dans le cas où elle n'est pas trouvée, il est nécessaire de fournir le fichier de paramètres d'entrée de RAMSES. Les deux autres sont lus à partir du fichier de configuration de la cosmologie fournit en entrée du code dynamique. De plus, comme dans le cadre des simulations cosmologiques, l'évolution du facteur d'échelle $a\,(t)$ doit être spécifiée au moyen de tables d'évolution. Pour cela, le programme prend en entrée le même fichier que celui qui est fourni au code dynamique. Il s'agit d'un fichier texte multicolonnes d'où il est possible d'extraire un tableau de valeurs de a et un tableau de valeurs du temps de regard en arrière t_{lookback}. L'objectif est d'obtenir à partir de cette table brute une nouvelle table à trois colonnes η, $a\,(\eta)$, $\frac{\mathrm{d}a(\eta)}{\eta}$, avec η le temps conforme et a le facteur d'échelle, qui est utilisée en interne du code de raytracing pour interpoler à toute position de la simulation la valeur du facteur d'échelle et/ou la valeur du paramètre de Hubble. En utilisant la valeur de h_0 pour les conversions d'unité et en intégrant t_{lookback}, on obtient facilement η en secondes.

Toutefois, les premiers tests de raytracing en métrique homogène basés sur cette table cosmologique conduisaient systématiquement à des écarts sur la distance de diamètre angulaire comparativement aux prévisions théoriques d'autant plus grands que l'intégration se poursuivait jusqu'à des redshifts élevés. Après de multiples tests, c'est la table elle-même qui a été identifiée comme origine de ces problèmes : en effet les valeurs de t_{lookback} et a du fichier d'évolution sont fournis avec une précision relative de 10^{-5}. Pour remédier à ce problème, la dérivée seconde de $a\,(\eta)$ est calculée numériquement au moyen d'une différentiation de Fornberg[139], lissée via un double filtre de Savitzky-Golay[140], puis réintégrée à deux reprises pour obtenir $\frac{\mathrm{d}a(\eta)}{\eta}$ et $a\,(\eta)$. La trajectoire d'un photon est ensuite intégrée pour une telle métrique homogène avec un pas d'intégration défini relativement aux paramètres de grille de la simulation dynamique destinée à être analysée. Cela fournit une référence pour réinterpoler la table à des valeurs optimales pour le raytracing en passe d'être effectué. À l'issue de cette procédure, on obtient une table compatible à 10^{-5} avec la table originale, lissée jusqu'à la dérivée seconde, et pour une gamme de valeurs déterminée à partir du maillage de la simulation à analyser. En procédant de la sorte, les écarts mentionnés en début de paragraphe disparaissent.

Acquisition de la distribution de potentiel

Une fois la cosmologie homogène acquise, vient le tour des perturbations, c'est-à-dire celui des résultats des simulations dynamiques en ce qui concerne la distribution de potentiel. Chaque tâche MPI hérite d'une portion de la sphère céleste à traiter, paramétrée par une direction (θ, ϕ) et un angle d'ouverture Θ. Le dossier dans lequel sont situés les fichiers issus de RAMSES [2] est d'abord parcouru pour déterminer leur nombre N_{files}. Un octree de niveau $\log_2\left(\sqrt[3]{N_{\text{files}}}\right)$ est ensuite reconstruit à partir des noms de fichiers pour aisément disposer de la localisation spatiale de chacun d'entre eux. L'en-tête de ces fichiers est utilisée pour vérifier le bon ordonnancement des dimensions : (x, y, z) ou (z, y, x) pour le calcul des indices monodimensionnels conduisant à leur indexation. Disposant de cet octree, il est aisé de déterminer la liste des fichiers à lire pour la portion d'espace à analyser par chaque tâche. Chacun des fichiers est chargé en mémoire, et les cellules dont l'intersection avec la zone de travail n'est pas nulle sont insérées à la suite des tableaux de clés et de valeurs constituant l'octree de données dont la pré-allocation avait été effectuée conformément au premier paragraphe de cette sous-section. Pour chaque cellule, six données physiques sont ainsi chargées dans l'arbre : $\left(\rho, \Phi, \frac{\mathrm{d}\Phi}{\mathrm{d}x}, \frac{\mathrm{d}\Phi}{\mathrm{d}y}, \frac{\mathrm{d}\Phi}{\mathrm{d}z}, a_{\text{cell}}\right)$ avec dans l'ordre la densité, le potentiel gravitationnel et ses dérivées et le facteur d'échelle d'extraction de la cellule. À l'issue de cette phase, les tableaux de clés et de valeurs sont triés conformément au chapitre précédent pour rétablir la cohérence de l'octree. Ce tri final global permet d'éviter une insertion en $\mathcal{O}(N)$ pour chacune des cellules dans un contexte qui ne nécessite pas encore d'effectuer des recherches dans l'arbre. Étant donné la durée potentiellement élevée de cette phase de lecture et de redistribution de cellules pour d'importants volumes de données, l'octree obtenu peut être optionnellement écrit tel quel dans des fichiers temporaires. Dans le cas où de multiples analyses sont envisagées, cela permet d'économiser un temps substantiel au redémarrage, chaque tâche se contentant alors de charger le fichier d'arbre qui la concerne. Avant de procéder au raytracing, un dernier traitement est effectué : on s'affranchit de la dépendance aux normalisations de RAMSES par une conversion des valeurs des grandeurs physiques en unités du système international [3]. Ce faisant, on garanti un maximum de clarté et de généricité à l'algorithme d'intégration dont l'application à d'autres sources de données pourrait être effectuée de façon transparente sans aucune modification.

Modes de calcul

Étant donné la nécessité de valider le code de multiples façons, quatre modes de calcul ont été implémentés, dont trois configurations de test en plus du raytracing standard. La sélection du mode de fonctionnement du programme est effectuée via le fichier de paramètres principal. Il est ainsi possible de vérifier la cohérence du maillage et de la structure du raffinement, de mettre le potentiel gravitationnel à 0 pour intégrer les géodésiques pour une métrique homogène tout en gardant le même maillage, et d'effectuer un test en métrique de Schwarzschild. Dans ce dernier cas, le programme peut faire l'économie du chargement des données de simulation puisqu'il s'agit d'étudier la trajectoire de la lumière dans une boîte au sein de laquelle une masse ponctuelle a été placée au centre.

Dans le cas standard, c'est-à-dire celui de l'intégration des géodésiques au sein de la simulation, un bouclage autour de certains paramètres est effectué, dans le cas où l'on souhaite explorer l'influence de quelques-uns des paramètres numériques et notamment le nombre de photons par bundle, l'angle d'ouverture initial des bundle, et la condition d'arrêt, ces trois éléments ayant été décrits au quatrième chapitre. Pour chacun des jeux de paramètres, $N_{\text{photons/bundle}} \times N_{\text{bundle}}$ trajectoires sont intégrées numériquement dans des directions aléatoires. À l'issue du calcul, trois possibilités existent : soit toutes les trajectoires sont enregistrées sur disque, soit seules les trajectoires des photons centraux de chaque bundle sont enregistrées, soit encore une réduction statistique des données est effectuée. Dans ce der-

2. Dans la pratique, il s'agit en fait des fichiers traités par le programme PFOF-SLICER pour redistribuer les données dans des cubes à partir de la courbe de Hilbert utilisée par RAMSES.
3. En double précision flottante et au-delà, le système international d'unités est parfaitement compatible avec l'intégrateur sans risquer une dénormalisation lors des calculs.

nier cas, la moyenne et l'écart-type des distances de diamètre angulaire sont calculées à chacun des redshifts accessibles dans la simulation : cela permet d'éviter la sauvegarde de chaque trajectoire individuelle si seules les propriétés statistiques de la distribution comptent.

Intégration des géodésiques

De nombreux détails relatifs à l'intégration des géodésiques ont déjà été donnés au chapitre quatre. On rappelle ici que le l'on cherche à intégrer le vecteur $\left(a, \eta, x, y, z, \frac{\mathrm{d}\eta}{\mathrm{d}\lambda}, \frac{\mathrm{d}x}{\mathrm{d}\lambda}, \frac{\mathrm{d}y}{\mathrm{d}\lambda}, \frac{\mathrm{d}z}{\mathrm{d}\lambda}\right)$ par rapport au paramètre affine λ suivant les équations géodésiques présentées au chapitre trois. Le facteur d'échelle est intégré en utilisant le paramètre de Hubble conforme $\frac{\mathrm{d}a}{\mathrm{d}\eta}$ fournit dans la table cosmologique précédemment décrite. Ce dernier est intégré avec les autres quantités pour maintenir la cohérence de l'ensemble du point de vue des erreurs numériques. À chaque pas de λ, le potentiel et ses dérivées sont interpolées dans la grille de simulation au moyen d'un Cloud-In-Cell inverse. Dans un cadre « cinématique », un intégrateur de Runge-Kutta d'ordre 4 suffit à obtenir les précisions requises, d'autant plus que le pas d'intégration adaptatif $\mathrm{d}\lambda$ évolue en fonction du niveau de raffinement de la grille. Cette caractéristique est spécifié via le paramètre $n_{\text{steps/cell}}$ fourni en entrée de l'algorithme. L'erreur numérique sur l'élément de longueur $\mathrm{d}s^2$ est évaluée au moyen du rapport de la partie spatiale sur la partie temporelle :

$$\epsilon = \Xi = \frac{\left(1 - 2\frac{\Phi}{c^2}\right)\left[\left(\frac{\mathrm{d}x}{\mathrm{d}\lambda}\right)^2 + \left(\frac{\mathrm{d}y}{\mathrm{d}\lambda}\right)^2 + \left(\frac{\mathrm{d}z}{\mathrm{d}\lambda}\right)^2\right]}{c^2\left(1 + 2\frac{\Phi}{c^2}\right)\left(\frac{\mathrm{d}\eta}{\mathrm{d}\lambda}\right)^2} \tag{7.1}$$

Pour chaque pas d'intégration, des quantités supplémentaires sont sauvegardées en prévision d'éventuelles analyses : le pas d'intégration $\mathrm{d}\lambda$, la densité ρ, le potentiel gravitationnel Φ et ses dérivées, le redshift calculé comme le rapport des métriques, et l'erreur ϵ. L'algorithme s'arrête dès que le potentiel ne peut plus être interpolé à la position (x,y,z), c'est-à-dire lorsque le photon entre dans une cellule de la surface de la zone spatiale non disponible pour la tâche MPI chargée de l'intégration. $N_{\text{photons/bundle}}$ trajectoires sont calculées de la sorte, ce qui permet ensuite, lors d'une seconde passe, de calculer l'évolution de la distance de diamètre angulaire, de la convergence, du cisaillement et de la magnification pour le bundle courant. Une fois terminé, ces données sont stockées en mémoire, et le bundle suivant est traité. Et ainsi de suite jusqu'à ce que les trajectoires de tous les bundles de photons aient été intégrées numériquement.

7.1.2 Architecture et implémentation

L'architecture logicielle déployée ici reprend tous les concepts présentés dans les chapitres précédents : utilisation d'une bibliothèque active, métaprogrammation par template, contrôle du processus d'instanciation via SFINAE, injection de code au moyen du CRTP... Le code est écrit en C++11 et se base sur l'utilisation de Magrathea ainsi que sur la mise en application de l'algorithme d'orthtree décrit au chapitre précédent. Une partie significative des performances et de la scalabilité atteintes est liée à ce dernier point.

La lecture des fichiers textes est effectuée au moyen d'un unique chargement en mémoire en tant que fichier binaire pour maximiser le débit, puis un transfert et une analyse ligne par ligne au moyen d'un `std::istringstream`. Le chargement des fichiers de données Ramses est, lui, légèrement plus complexe. L'intégralité de chaque fichier à lire est chargée sur le nœud dans un buffer de `char` au moyen des routines fournies par Magrathea. Le contenu des cellules est ensuite traité, sachant qu'il est originellement encodé au format Fortran, c'est-à-dire que chaque écriture est précédée et suivie d'un « record marker » de 4 octets indiquant la taille en octets de l'enregistrement en question. Ainsi, par exemple, les valeurs du potentiel dans un fichier de N_{cells} cellules en simple précision se présente sous la forme d'un tableau de N_{cells} flottants précédé et suivi d'un entier de 4 octets de valeur $N_{\text{cells}} \times$ `sizeof(float)`. Ce format simple permet d'ajouter une sécurité quant à la bonne lecture de

fichiers binaires. RAMSES travaillant niveau par niveau, les tableaux de cellules sont enregistrés de la même manière : il est donc nécessaire d'itérer sur les niveaux pour effectuer la lecture complète d'un fichier.

Chaque cellule est analysée individuellement en vue de déterminer, si, oui ou non, elle appartient à la zone spatiale traitée par le processus MPI courant. Dans le cas positif, les données sont insérées dans l'arbre en construction au moyen d'un `emplace_back`. Comme, dans la pratique, il s'agit de traiter le Full Universe Run, des indices de 128 bits `__uint128_t` ont été choisis pour être à même de stocker les $13 + 6 = 19$ niveaux de raffinement de la simulation. Le contenu de chaque cellule, lui, est représenté par un « tuple-like » eulérien de six flottants simple précision, les enregistrements du Full Universe Run ayant fait l'économie de la double précision pour des problématiques d'archivage : $\left(\rho, \Phi, \frac{\mathrm{d}\Phi}{\mathrm{d}x}, \frac{\mathrm{d}\Phi}{\mathrm{d}y}, \frac{\mathrm{d}\Phi}{\mathrm{d}z}, a_{\text{cell}}\right)$ où a_{cell} correspond au facteur d'échelle d'extraction de la cellule et qui permet d'identifier les cellules appartenant à une même coquille. La construction de l'octree nécessite bien entendu le calcul des indices des cellules au fur et à mesure de leur insertion. Comme cela a été déjà précisé, l'espace mémoire nécessaire au stockage de l'arbre est pré-alloué par des `std::vector::reserve` en vue d'éviter les réallocations. À l'issue du chargement, l'arbre est trié une fois pour toutes par un `std::sort`, qui correspond, dans le cas de l'implémentation de la librairie standard de GCC, à un introsort.

Enfin, au niveau du calcul des trajectoires, le tirage aléatoire de la direction initiale de propagation est réalisée au moyen d'un Mersenne Twister[141] 32-bits de période $2^{19937} - 1$ initialisé soit par une graine fournie par l'utilisateur, soit par une graine générée par le système, de façon non-déterministe si possible, au moyen d'un `std::random_device`. Concernant l'intégration elle-même, tant l'interpolation tridimmensionnelle par CIC inverse que l'intégration de Runge-Kutta ont été implémentées de façon générique. C'est-à-dire que, quel que soit le contenu des cellules de l'arbre, pour une position (x, y, z) fournie en entrée, toutes les composantes interpolables, c'est-à-dire, toutes les composantes arithmétiques du tuple-like eulérien, sont interpolées récursivement. Autrement dit, pour chaque position (x, y, z) passée à l'interpolateur, ce dernier produit un ensemble $\left(\rho, \Phi, \frac{\mathrm{d}\Phi}{\mathrm{d}x}, \frac{\mathrm{d}\Phi}{\mathrm{d}y}, \frac{\mathrm{d}\Phi}{\mathrm{d}z}, a_{\text{cell}}\right)$ calculée à cette position en fonction des cellules environnantes. Dans le cas où les cellules comprendraient une composante non-arithmétique, cette dernière serait ignorée pour l'interpolation. L'implémentation de l'intégrateur de Runge-Kutta, elle, permet de spécifier de tableau de Butcher à la compilation, c'est-à-dire l'ordre et les cœfficients utilisés. De manière analogue au CIC, le vecteur d'intégration reprend le concept de « tuple-like » si bien que l'algorithme s'adapte automatiquement à tout type contenu. La seule chose qu'il est nécessaire de spécifier est la façon dont est intégré le vecteur, c'est-à-dire en l'occurrence la forme des équations géodésiques présentées au quatrième chapitre.

7.1.3 Parallélisation

Parallélisation intra-nœuds

Comme mentionné précédemment, le programme fait l'objet d'une parallélisation hybrique MPI / `std::thread`. Le nombre de tâches du niveau MPI est spécifié au lancement du programme tandis que le nombre de threads disponible est récupéré automatiquement en interne via la fonction statique `std::thread::hardware_concurrency`. Le threading intervient à de multiples reprises dans l'algorithme. Toutefois, deux phases tirent particulièrement parti de celui-ci : le chargement des données, et l'intégration des géodésiques. Dans le premier cas, pour maximiser la vitesse de transfert, on laisse un seul thread se charger de transférer les données du fichier courant des disques vers la mémoire locale du nœud. On évite ainsi du même coup de surcharger le système de fichiers par trop d'accès simultanés au niveau global. Pour chaque niveau de raffinement, les tableaux contenant les données des cellules sont donc lus par le thread principal. Une fois en mémoire, N_{threads} sont lancés pour détecter si oui ou non, chaque cellule doit être traitée par le nœud en fonction de sa position. Un tableau d'entiers de taille N_{cells} est alors rempli de façon parallèle en spécifiant 0 ou 1 suivant que la cellule

doit être ajoutée à l'arbre ou non. À l'issue de cette procédure, le tableau d'entiers est parcouru par le thread principal et les 1 sont remplacés par leur somme cumulative commencée à 1. Cela permet de produire une liste d'indices pour préparer le transfert des données : par exemple, un 10 signifie que le contenu de la cellule à cet emplacement est la dixième à faire l'objet d'un transfert dans l'arbre. Ainsi, on commence par redimensionner les tableaux dynamiques liés à l'arbre, et le transfert des cellules des tableaux originaux vers l'octree peut ensuite être effectué de façon parallèle car chaque thread connait *a priori* l'emplacement mémoire de destination. Une fois l'arbre final produit, la conversion des unités des quantités physiques des cellules peut être effectué, lui aussi, de façon parallèle : il suffit de remplacer un `std::for_each` par un `parallel_for_each` tel que celui fourni par MAGRATHEA. La seconde phase au cours de laquelle le threading a un rôle important a joué est celle de l'intégration des géodésiques. Le parallélisme est trivial puisqu'il s'agit d'intégrer les trajectoires de N_{bundle} différents et indépendants. De même que précédemment, une parallélisation de boucle `parallel_for_each` permet d'atteindre une scalabilité quasi parfaite en occupant de façon optimale tous les threads d'une tâche MPI.

Parallélisation inter-nœuds

La parallélisation inter-nœuds, elle, intervient dans le découpage spatial de la simulation. Il s'agit de répartir équitablement la sphère reconstruite à partir des coquilles sur les N_{tasks} disponibles. Les fichiers bruts résultant d'un découpage de la boîte de simulation en cubes, il serait naturel d'opter pour un découpage du même genre, chaque tâche chargeant le même nombre de fichiers. Cela pose trois problèmes. Tout d'abord le poids des fichiers n'est pas le même : ceux situés au centre de la sphère, autour de l'observateur, correspondent à des redshifts faibles, très structurés avec de nombreux éléments de grille, alors que ceux situés à la surface de la sphère, ou au-delà ont peu ou pas d'éléments de grille. Ensuite, la charge de calcul est très inhomogène puisque les tâches héritant des huit cubes centraux devront de charger des premières étapes d'intégration de tous les photons. Enfin, et c'est là l'aspect principal : dans un cadre cinématique, les données de la grille n'évoluent pas. Autrement dit, la grille est fixée une fois pour toutes. Comme en régime de faibles perturbations les déflexions sont faibles et les trajectoires des photons sont essentiellement radiales, un découpage spatial intelligent devrait permettre une parallélisation triviale, c'est-à-dire une parallélisation sans communications ou avec un nombre de communications minimal. Or un découpage cartésien nécessite de communiquer les photons de tâche en tâche à mesure qu'ils passent de cube en cube. Pour régler ce problème, une parallélisation en cônes à base circulaire a été choisie. En effet, en découpant la sphère en zones spatiales coniques, l'intégralité de chaque trajectoire peut être traitée par une unique tâche. Se pose alors deux questions : comment traiter la zone de départ des photons autour de l'observateur, et comment déterminer le découpage en cônes ?

La réponse à la première question est simple : il suffit d'adjoindre au cône une sphère autour de l'observateur de manière à ce que l'interpolation CIC inverse ne pose pas de problème sur les premiers pas de temps. Dans la pratique, il semble que pour une simulation de 8192^3 éléments de grille, une sphère centrale de 32 cellules de rayon est bien adaptée et permet d'éviter tout problème d'interpolation avec une empreinte mémoire faible, puisqu'il s'agit d'un peu moins de 140 000 cellules coarse, ce qui est, au regard de la mémoire disponible, faible. Au final, toutes les tâches MPI disposent donc, en plus de leur cône, d'une zone spatiale sphérique autour de l'observateur.

Reste la question principale à savoir celle de la détermination des paramètres des cônes pour paver la sphère. En fait, cela correspond à trouver une solution au problème de Tammes[142], puisqu'il s'agit de trouver la disposition optimale de N_{tasks} points sur la sphère afin que la distance minimale entre deux points soit la plus grande possible, ces points représentant les centres des bases circulaires des cônes. En d'autres termes, on cherche à paver la sphère tridimensionnelle par N_{tasks} cercles avec recouvrement, où chacun de ces cercles correspond à la base d'un cône. Le problème peut en réalité être découpé en deux étapes : le premier qui est de distribuer les points sur la sphère, et le second qui

est de déterminer l'angle d'ouverture de cônes. Pour le premier aspect, le problème de Tammes est un problème mathématique ouvert, et le résoudre n'est pas le sujet de cette thèse. Il n'existe pas de solution connue pour un nombre de points arbitraire, même si des solutions exactes existent pour quelques N faible. Dans notre cas, comme N_{tasks} peut être très élevé, on utilise une solution approchée. Cette solution consiste à « enrouler » une spirale autour de la sphère et à distribuer les points le long de cette spirale :

```
1  std::vector<std::array<double, 3>> distribute(std::size_t n)
2  {
3    const double cst = 8*std::atan(1)*(2-((1+std::sqrt(5))/2.));
4    const double step = 2./n;
5    std::array<double, 3> tmp{};
6    std::vector<std::array<double, 3>> xyz(n);
7    for (std::size_t i = 0; i < n; ++i) {
8      tmp[0] = i*step-1+step/2;
9      tmp[1] = std::sqrt(1-tmp[0]*tmp[0]);
10     tmp[2] = i*cst;
11     xyx[i][0] = std::cos(tmp[2])*tmp[1];
12     xyy[i][1] = tmp[0];
13     xyz[i][2] = std::sin(tmp[2])*tmp[1];
14   }
15   return xyz;
16 }
```

Les résultats pour quelques valeurs de N de cette méthode sont illustrés en figure 69 : plus N est grand plus la solution s'approche du cas idéal. Pour un nombre de points très faible, c'est-à-dire typiquement en deçà de 8, la solution n'est cependant pas optimale. Les solutions exactes étant connues jusqu'à $N = 6$, il serait possible de les encoder telles quelles dans le programme en lieu et place de la méthode approchée pour N petit.

Le second aspect du problème, lui, concerne l'angle d'ouverture des cônes une fois que la position des centres des bases a été fixée. L'objectif est de paver toute la sphère, en minimisant les recouvrements : un angle d'ouverture trop petit laisserait des zones non attribuées à des tâches, tandis qu'un angle d'ouverture trop grand conduirait à un besoin de mémoire trop élevé. Pour résoudre ce problème, on utilise une approche numérique. Et dans, ce cadre il apparait que le demi-angle d'ouverture :

$$\frac{\Theta}{2} = \sqrt{\pi} \sin^{-1}\left(\frac{d_{min}}{D}\right) \tag{7.2}$$

où d_{min} est la distance minimale entre deux points sur la sphère, et où D est le diamètre de cette dernière, semble être quasi-optimal, du moins pour le nombre de tâches testé. Il ne s'agit toutefois en aucun cas d'un résultat analytique : il s'agit juste du résultat d'une exploration numérique.

En utilisant cette procédure, les différentes tâches MPI peuvent donc se répartir le travail et charger les fichiers et les cellules qui intersectent leur cône. Avant de procéder à la lecture, chaque tâche précalcule une liste de fichiers à lire. Pour éviter que l'ordre d'accès soit le même pour toutes les tâches, et donc pour éviter un engorgement lors de la lecture, la liste est triée dans un ordre aléatoire. Dans le cadre du code développé ici, le choix d'un nombre de communications minimal a été effectué au prix d'un accès aux fichiers plus intense. Il aurait été possible de répartir la lecture équitablement sur toutes les tâches et de procéder à une redistribution des cellules au moyen de communications beaucoup plus intenses. Il conviendrait de tester cette approche pour en comparer les performances. Notons qu'une fois que la structure de l'arbre local de chaque tâche a été écrit dans un fichier temporaire, il n'est plus nécessaire d'effectuer aucune communication hormis celles liées aux réductions statistiques lorsqu'il s'agit de calculer des moyennes ou des écart-types de grandeurs physiques sur tout le ciel. Pour

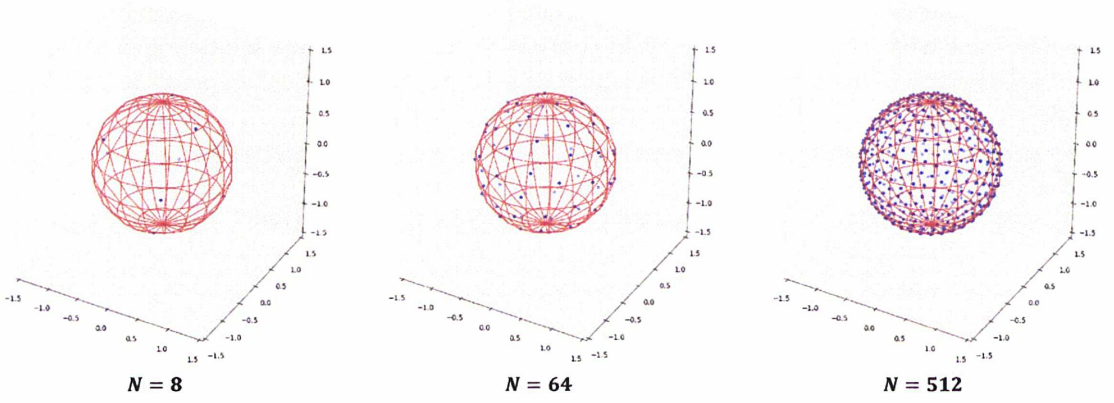


FIGURE 69 – Approximation de la solution au problème de Tammes par l'enroulement d'une spirale sur une sphère. Pour répartir le volume de la sphère céleste sur N tâches MPI, on procède à un découpage en cônes à base circulaire. Obtenir un découpage équilibré correspond à placer les points correspondants aux centres de la base des cônes à des positions solution du problème de Tammes, telle que la distance minimale entre deux points soit la plus grande possible. Il est possible d'obtenir une solution approchée à ce problème en disposant les points sur une spirale enroulée autour de la sphère. Plus le nombre de points est élevé, plus la solution est pertinente comme on peut le constater qualitativement sur les figures ci-dessus.

terminer, un dernier problème se présente : comment éviter d'obtenir une densité de photons plus importante dans les zones de recouvrement des cônes ? Pour cela, les générateurs aléatoires de tous les processus MPI sont initialisés avec la même graine, et tous les processus tirent des photons dans toutes les directions. Ne sont gardés localement que les photons pour lesquels la projection de la direction initiale sur l'axe du cône local est maximale comparativement aux autres cônes. Ainsi, il est possible de déterminer de manière déterministe le processus reponsable de l'intégration de la trajectoire de chaque photon, y compris dans les zones de recouvrement. Là, encore, plutôt que d'initialiser localement tous les photons pour en rejeter l'immense majorité il serait possible de paralléliser le problème au moyen d'un échange de photons entre processus. Mais la génération aléatoire des directions est extrêmement rapide, et la solution sans communications s'avère être au final plus performante. Cette différence de performances devrait toutefois se réduire et s'inverser pour un nombre de tâches très élevé.

7.2 Tests et vérifications

7.2.1 Construction de l'arbre

Le premier élément à vérifier pour contrôler le bon fonctionnement de l'algorithme, est l'arbre lui-même. En effet, on se place dans toute la suite dans une démarche de vérification en vue de garantir la robustesse des résultats physiques obtenus à l'issue du raytracing. La validation de la structure de l'octree permet de s'assurer que les données sur lesquelles il va s'agir de travailler sont bien importées. Pour cela, plusieurs tests sont réalisés automatiquement par le code lorsque le mode de test de la grille est enclenché. Le premier élément calculé est le nombre de grilles par niveaux, le niveau minimal, et le niveau maximal. Le volume total du niveau grossier, et le volume total des cellules feuilles, c'est-à-dire des cellules non raffinées, sont ensuite comparées : ces derniers doivent être rigoureusement égaux. Au niveau de la structure de l'arbre lui-même, on s'assure de la présence systématique de 8 cellules filles pour les raffinements, et que chaque cellule non grossière dispose d'un parent du niveau juste supérieur. Concernant les données physiques, on valide la cohérence des valeurs entre les différents niveaux de raffinement, entre les parents et les enfants, sachant que la version de RAMSES utilisée se base sur une approche multigrille. On s'assure également du bon fonctionnement du Cloud-In-Cell inverse et de la cohérence des valeurs du potentiel et de la densité.

Cette batterie de tests a permis de mettre en évidence une série de subtilités existantes dans la toute première version du code, qui ont été résolues par la suite. L'élément inattendu fut le caractère creux de la grille, alors que RAMSES raffine systématiquement en huit cellules filles. Cela était lié à l'algorithme d'extraction des coquilles pour la construction du cône de lumière : en effet, dans le cas où une cellule est à la frontière d'une coquille, et que cette cellule se raffine au moment où elle est extraite, il se peut que les huit fils ne soient pas sélectionnés et enregistrés. Dans le cas de la grille du Full Universe Run, il s'agissait de moins de 0.01% des cellules. Plusieurs solutions pour obtenir une grille complète étaient disponibles : soit créer les cellules filles manquantes en raffinant artificiellement leur parent et en interpolant la valeur des données physiques par Cloud-In-Cell inverse, soit supprimer les raffinements des cellules concernées. Les deux options ont été implémentées dans le code, et c'est la première qui a été utilisée, une fois la vérification faite que les résultats d'intégration des trajectoires étaient semblables. Hormis ces problèmes originaux, corrigés depuis, les autres aspects de la grille avaient été validés avec succès.

7.2.2 Intégration dans le cas homogène

Pour tester l'intégrateur, il est intéressant de se reporter au cas le plus simple : celui du potentiel Φ nul, c'est-à-dire au cas homogène, en expansion, sans perturbation. Dans ce mode de test, la grille et les données de la simulation sont bien lues, mais le potentiel et ses dérivées sont ensuite artificiellement fixés à zéro. Cela permet de se rapprocher du cas d'intégration réel pour lequel le code dispose d'une grille raffinée par endroits. Pour tester la précision relative de l'algorithme, il suffit de comparer deux redshifts : celui issu de l'intégration et calculé comme le rapport des métriques, et celui, analytique, obtenu à partir du facteur d'échelle déduit de la distance comobile à l'observateur $a(\chi)$. La figure 70 illustre la variation de l'erreur relative sur le redshift en fonction du nombre de pas d'intégration, pour un calcul effectué avec un algorithme de Runge-Kutta d'ordre 4, et ce, à différent redshifts. Le fait que l'erreur augmente avec le redshift est simplement lié à l'accumulation de l'erreur le long de la trajectoire du photon. La pente obtenue dans le régime où les erreurs sont supérieures à 10^{-12} correspond bien à la pente attendue pour un intégrateur d'ordre 4. Étant donné que la profondeur des cônes du Full Universe Run est de $z \approx 30$ et que l'on se fixe comme seuil d'erreur relative 10^{-12} en double précision, la figure 70 nous indique que le choix de 4 étapes d'intégration par cellule coarse est optimal pour une simulation 8192^3 de la taille de l'Univers Observable.

Il est également possible d'analyser l'évolution de l'erreur directement en fonction du redshift, ou de façon équivalente en fonction du temps de regard en arrière, et ce, pour différents nombres de pas d'intégration par cellule. Les résultats sont présentés en figure 71 et 72 pour un intégrateur d'Euler et pour un intégrateur de Runge-Kutta d'ordre 4. L'avantage du second intégrateur sur le premier en termes d'erreur relative est mis en évidence : à 4 pas d'intégration par cellule, et à $z = 30$, l'erreur relative sur le redshift est de l'ordre de 3×10^{-3} dans le premier cas, tandis qu'elle est de l'ordre de 3×10^{-13} dans le second. C'est donc sans hésitation que le second sera choisi pour le raytracing. D'autre part, cette représentation permet de vérifier l'évolution de l'accumulation des erreurs. Dans le cas de la méthode d'Euler, l'erreur ne cesse d'augmenter à mesure que l'on s'écarte de l'observateur, alors que dans le cas de la méthode de Runge-Kutta d'ordre 4 elle reste à peu près constante jusqu'à $z = 5$ avec des valeurs typiquement comprises entre 10^{-15} et 2×10^{-14}. C'est-à-dire que dans ce cas, l'erreur relative est contrainte par le « machine epsilon » de la double précision. La figure 72 permet également de comparer l'avantage de disposer de 8 pas d'intégration par cellule au lieu de 4. Dans les domaines de redshift d'intérêt, force est de constater que cet avantage reste faible. C'est pourquoi 4 pas d'intégration par cellule seront utilisés dans le cadre du raytracing au sein du Full Universe Run. L'intégration dans le cas homogène permet donc de s'assurer du bon fonctionnement de l'intégrateur, mais également de quantifier les erreurs d'intégration et de calibrer le nombre de pas par cellule nécessaire pour atteindre une bonne précision sur le calcul des quantités physiques.

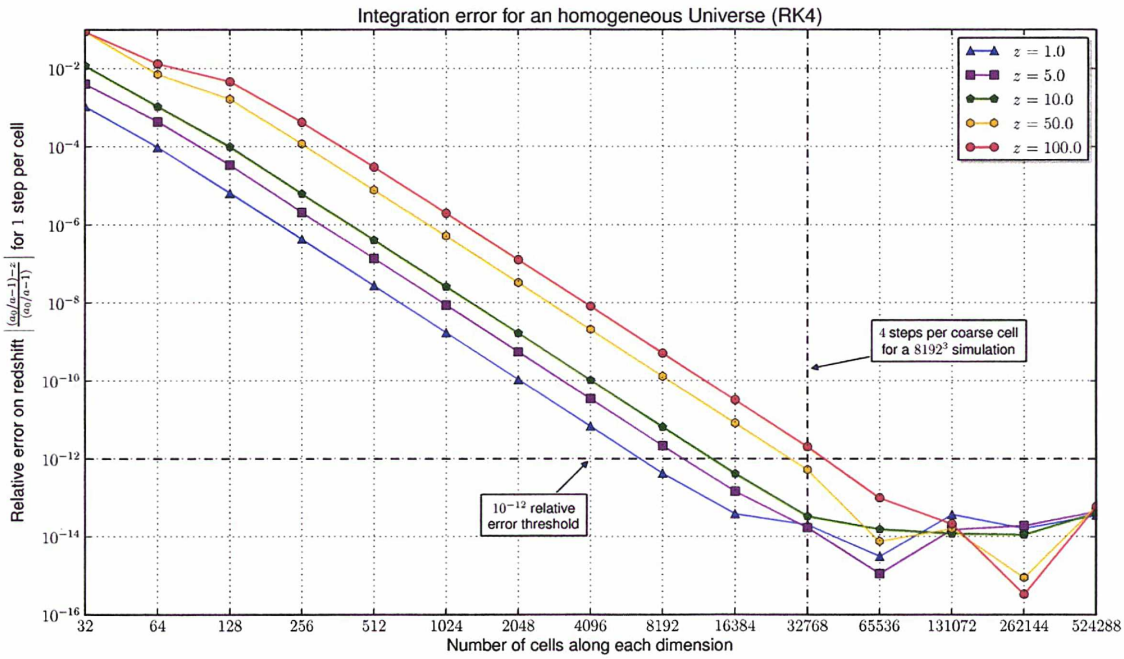


FIGURE 70 – Évolution de l'erreur relative sur le redshift en fonction du nombre de pas d'intégration dans le cas homogène pour un intégrateur d'ordre 4. L'évolution de l'erreur relative en fonction du nombre de pas d'intégration est aussi dépendante du redshift du fait de l'accumulation des imprécisions. La pente des courbes dans le régime au-dessus de 10^{-12} est caractéristique d'un intégrateur d'ordre 4. Dans le cadre du Full Universe Run, on cherche à garantir une erreur relative inférieure à 10^{-12} en double précision jusqu'à $z = 30$. La figure nous indique que 32768 étapes d'intégration, c'est-à-dire 4 pas par cellule coarse, permettent de répondre à ce critère.

7.2.3 Intégration en métrique de Schwarzschild

Principe et solution analytique

La grille et l'intégrateur ayant été testés, il est maintenant nécessaire de valider la cohérence physique des quantités calculées lorsque le potentiel et ses dérivées ne sont pas nuls. Pour cela, la métrique de Schwarzschild est idéale puisqu'il est possible d'obtenir des solutions analytiques au problème. Étant donné l'utilité de cette configuration, une attention particulière va être portée sur la dérivation des solutions analytiques du problème compatibles avec les simulations. Ces solutions analytiques pourront ensuite être exploitées comme cas de référence dans le régime des simulations cosmologiques. Quelques mesures numériques ont été effectuées, mais elles nécessiteraient, à terme, une étude plus exhaustive. En effet, c'est avant tout à la présentation d'une méthode rigoureuse d'évaluation des erreurs numériques, plus qu'à la présentation de mesures qu'est destinée cette partie.

Le principe est le suivant : il va s'agir d'évaluer la déflection d'un photon provenant de l'infini avec un paramètre d'impact b par une masse sphérique M de rayon r_s comme illustré en figure 73. Cet exercice va naturellement nécessiter de calculer avec justesse, ou tout du moins avec la plus grande précision possible [4], la trajectoire du photon, et permettra d'évaluer deux composantes essentielles du raytracing :

- la précision de l'intégrateur
- l'influence de la discrétisation du potentiel sur la grille

4. Idéalement avec une erreur de l'ordre de $\varepsilon \approx 10^{-15}$ en double précision.

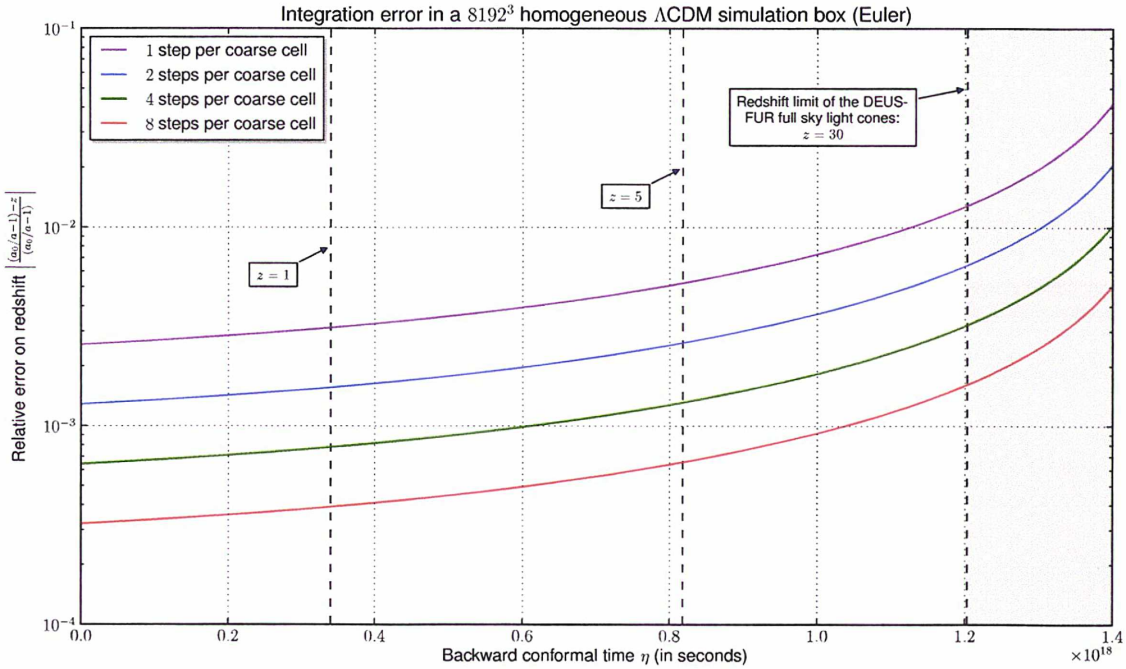


FIGURE 71 – Évolution de l'erreur relative sur le redshift en fonction du temps de regard en arrière conforme pour un intégrateur d'Euler. L'erreur relative obtenue par la méthode d'Euler pour 8 pas d'intégration par cellule et moins est très élevée puisqu'elle est dans tous les cas supérieure à 3×10^{-4}, ce qui rend cette méthode inutilisable pour les analyses de raytracing, à moins d'augmenter considérablement le nombre de pas d'intégration. De plus, cette dernière augmente à mesure que l'on s'écarte de l'observateur, signe que les erreurs s'accumulent.

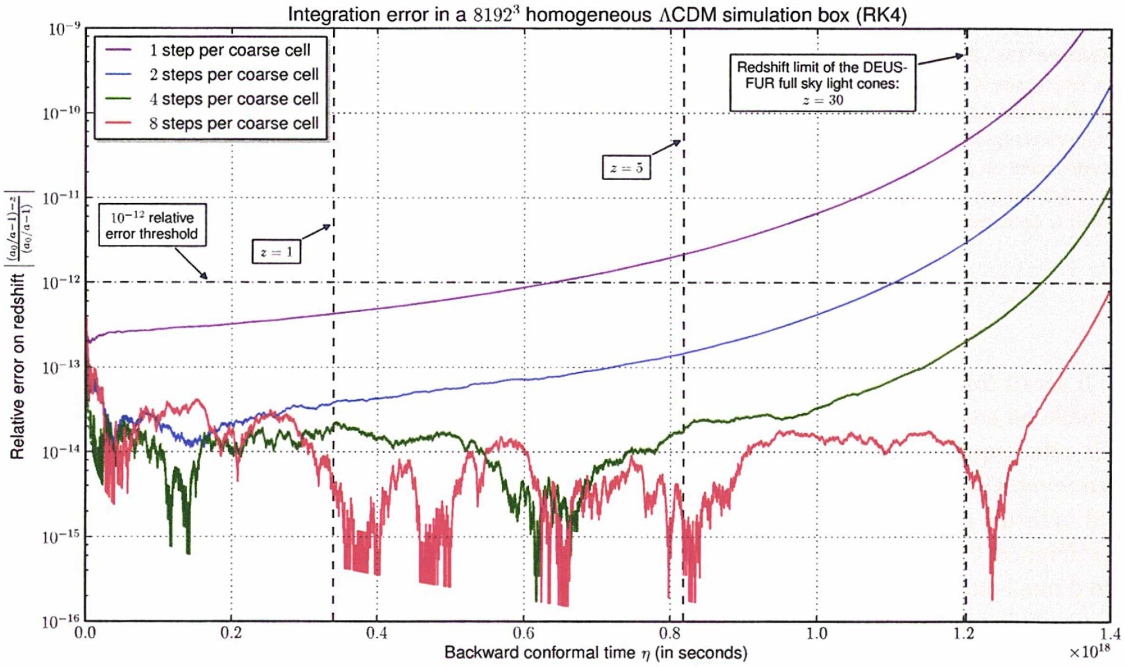


FIGURE 72 – Évolution de l'erreur relative sur le redshift en fonction du temps de regard en arrière conforme pour un intégrateur de Runge-Kutta d'ordre 4. Comparativement à la figure 71, les précisions atteintes par la méthode de Runge-Kutta d'ordre 4 sont bien supérieure. Ces dernières semblent d'ailleurs limitées par la double précision jusqu'à $z = 5$ pour 4 pas d'intégration par cellule. L'avantage de disposer de 8 pas par cellule dans la gamme de redshifts étudié n'est pas évident. C'est donc $n_{\text{steps/cell}} = 4$ qui sera choisi dans la suite. On vérifie encore une fois que l'erreur à $z = 30$ est inférieure 10^{-12}.

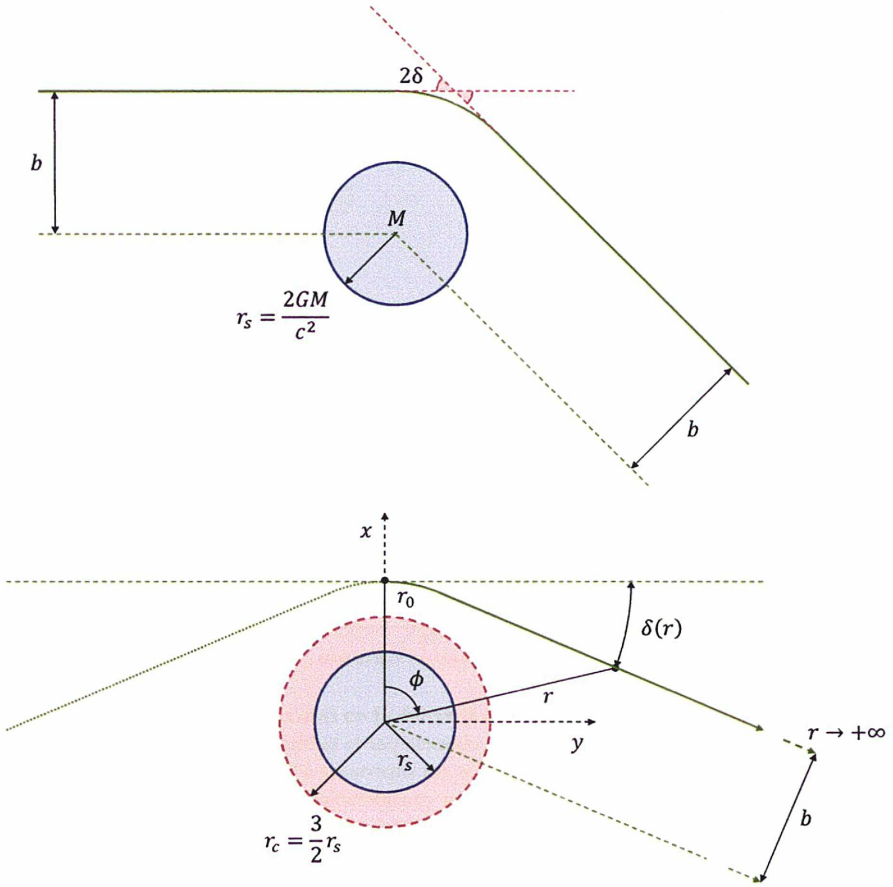


Figure 73 – Principe de validation de l'intégrateur en métrique de Schwarzschild. L'objectif est de comparer la trajectoire d'un photon provenant de l'infini avec un paramètre d'impact b et notamment la valeur de l'angle total de déflexion 2δ dans le cadre d'une résolution « analytique » du problème et dans le cadre de l'intégrateur utilisé pour le raytracing comme illustré dans la partie supérieure de la figure. Dans la partie inférieure, on présente une version symétrisée du problème où l'on considère de façon équivalente une particule sans masse « tirée » du point $(r_0, 0)$ et s'éloignant asymptotiquement vers le paramètre d'impact b. Pour tout point de la trajectoire $\phi(r)$, il est possible de définir un demi-angle de déflexion $\delta(r)$.

Il serait tout à fait envisageable d'établir le cas de référence par intégration de l'équation différentielle en coordonnées polaires satisfaite par la trajectoire du photon dans le plan de propagation, mais cela risquerait d'introduire le même type d'erreurs numériques que celles inhérentes au code de raytracing. Une solution alternative consisterait à n'évaluer l'erreur qu'au moyen de la déflexion 2δ dont la forme analytique est bien connue et qui ne nécessite pas d'intégration numérique dans le cas de référence. Cependant, comme il est impossible d'effectuer la propagation d'un rayon lumineux au sein d'une boîte de simulation infiniment grande, cette quantité est inaccessible au code de raytracing. Et quand bien même il serait possible d'exécuter le code pour des boîtes extrêmement grandes, accéder à 99% voire 99.9% de la déflexion totale n'aurait que peu d'intérêt pour mesurer une erreur relative que l'on espère largement inférieure à 10^{-3}. Pour ces raisons, établir une forme analytique exacte des géodésiques nulles dans une métrique de Schwarzschild est un prérequis nécessaire à la détermination de l'erreur numérique que l'on cherche à effectuer.

La dérivation d'une telle forme est détaillée en annexe. À l'issue du calcul, on obtient la formule

suivante en fonctions de Weierstrass :

$$\boxed{\phi(r) = 2\wp^{-1}\left(\alpha - \frac{1}{3} + \frac{3\alpha(\alpha-2)}{\frac{r_s}{r} - \alpha}\right) \qquad \text{avec} \qquad \alpha = \frac{r_s}{r_0}} \tag{7.3}$$

avec r et ϕ les coordonnées polaires du photon précisées en figure 73, r_s le rayon de Schwarzschild et r_0 la distance d'approche maximale calculée en annexe et qui vaut :

$$\boxed{r_0 = \frac{2b}{\sqrt{3}}\cos\left(\frac{1}{3}\cos^{-1}\left(-\frac{3\sqrt{3}r_s}{2b}\right)\right)} \tag{7.4}$$

avec b le paramètre d'impact.

Détermination numérique du cas de référence

À partir de l'expression de $\phi(r)$ 7.3 il est assez aisé de calculer les valeurs de la déviation avec une précision maximale. Pour cela, la méthode la plus simple consiste d'abord à implémenter $\wp$ puis a l'inverser. Pour la première étape, on exprime $\wp$ comme :

$$\wp(z) = e_3 + \frac{e_1 - e_3}{\mathrm{sn}^2(w,k)} \qquad \text{avec} \qquad \begin{cases} w = z\sqrt{e_1 - e_3} \\ k = \sqrt{\frac{e_2 - e_3}{e_1 - e_3}} \end{cases} \tag{7.5}$$

où les expressions de (e_1, e_2, e_3) sont données en annexe et où sn est la fonction sinus de Jacobi :

$$\mathrm{sn}(u,k) = \sin(\psi) \qquad \text{pour} \qquad u = \int_0^{\psi} \frac{\mathrm{d}\theta}{\sqrt{1 - k^2\sin(\theta)}} \qquad \text{avec} \qquad 0 \leq k < 1 \tag{7.6}$$

dont un exemple d'implémentation est fourni en figure 74. Le résultat de cette fonction peut être calculé à la précision machine en quelques itérations seulement. Une fois $\wp$ implémenté, se pose la question de son inversion pour calculer ϕ. Pour cela, on commence par déterminer les demi-périodes fondamentales de $\wp$:

$$\omega_1 = \frac{1}{\sqrt{e_1 - e_3}} K\left(\sqrt{\frac{e_2 - e_3}{e_1 - e_3}}\right) \qquad \text{et} \qquad \omega_3 = \frac{i}{\sqrt{e_1 - e_3}} K\left(\sqrt{\frac{e_1 - e_2}{e_1 - e_3}}\right) \tag{7.7}$$

avec $K(k) = F\left(\frac{\pi}{2}, k\right)$. L'inversion de $\wp$ sur $]0, \omega_1]$ peut ensuite être implémenté comme une simple recherche binaire sur des flottants, le temps de calcul n'étant pas critique dans le cas présent.

A ce stade, toutes les quantités d'intérêt pour le cas de référence peuvent être évaluées avec une grande précision, et notamment, en complément de 7.3 :

$$\boxed{r(\phi) = \frac{3r_s\left(\wp\left(\frac{\phi}{2}\right) - \beta\right)}{(3\beta+1)\left(\wp\left(\frac{\phi}{2}\right) + 2\beta - 1\right)} = \frac{r_s\left(\wp\left(\frac{\phi}{2}\right) - \alpha + \frac{1}{3}\right)}{\alpha\left(\wp\left(\frac{\phi}{2}\right) + 2\alpha - \frac{5}{3}\right)}} \tag{7.8}$$

ainsi que le demi-angle de déflexion :

$$\boxed{\delta(r) = \phi(r) - \cos^{-1}\left(\frac{r_0}{r}\right) = 2\wp^{-1}\left(\alpha - \frac{1}{3} + \frac{3\alpha(\alpha-2)}{\frac{r_s}{r} - \alpha}\right) - \cos^{-1}\left(\frac{r_0}{r}\right)} \tag{7.9}$$

Une illustration du résultat obtenu dans le cas d'une masse $M = M_\odot$ est donné en figure 75. On notera notamment que pour $b = R_\odot$, une boîte de $100b$ de côté ne permettrait d'obtenir qu'environ 99% de la valeur finale δ_∞ via du raytracing, confirmant, de fait, la nécessité de disposer de la déflexion le long de toute la trajectoire avec exactitude. Dans la pratique, les méthodes décrites dans les paragraphes précédents permettent d'atteindre des erreurs relatives de l'ordre de 10^{-14} sur $\delta(r)$, fixant ainsi avec précision le cas de référence pour différentes masses M et pour différents paramètres d'impacts b.

C++11 code: computation of the Jacobi elliptic function sn

```cpp
// Computation of the Jacobi elliptic function sn
template <typename T,
          int N = 2*sizeof(T),
          class = typename std::enable_if<
          std::is_floating_point<T>::value>::type,
          class = typename std::enable_if<
          (N >= 2*sizeof(float))>::type>
T sn(T u, T k)
{
  // Initialization
  using limits = std::numeric_limits<T>;
  static constexpr T one = 1;
  static constexpr T epsilon = limits::epsilon();
  int n = N;
  T x = one, y = T(), sn = T(), cn = T(), dn = T();
  T onemk2 = k*k < one ? one-k*k
           : throw std::domain_error("1 <= k*k");
  std::array<T, N+N> coeff = {{}};

  // No approximation for standard values of k
  if (std::isnormal(k)) {

    // Computation of coefficients
    for (int i = 0; i < n; ++i) {
      coeff[i] = x;
      onemk2 = std::sqrt(onemk2);
      coeff[N+i] = onemk2;
      y = (x+onemk2)/(one+one);
      if (std::abs(x-onemk2) > epsilon*x) {
        onemk2 *= x;
        x = y;
      } else {
        n = ++i;
        i = n;
      }
    }
    u *= y;
    sn = std::sin(u);
    cn = std::cos(u);
    dn = one;

    // Computation of sn if normal
    if (std::isnormal(sn)) {
      x = cn/sn;
      y *= x;
      for (int i = n-1; i >= 0; --i) {
        x *= y;
        y *= dn;
        dn = (coeff[N+i]+x)/(coeff[i]+x);
        x = y/coeff[i];
      }
      x = one/std::sqrt(y*y+one);
      sn *= x/std::abs(sn);
      cn = y*sn;
    }

  // Approximation for small values of k
  } else {
    sn = std::tanh(u);
    cn = one/std::cosh(u);
    dn = cn;
  }

  // Final result
  return sn;
}
```

FIGURE 74 – Exemple d'implémentation de la fonction sinus de Jacobi. L'algorithme est inspiré de celui fourni par Press et al. 2007[143]. Dans le cas où le module k est proche de zéro, le résultat est approximé en termes de fonctions hyperboliques, sinon l'algorithme converge vers la bonne valeur en prenant en compte la précision machine.

Précision du raytracing en métrique de Schwarzschild

Le cas de référence ayant été déterminé en minimisant, autant que faire se peut, les erreurs numériques, la précision du raytracing peut maintenant être évaluée. Pour cela plusieurs sources d'erreurs sont à prendre en compte. La première d'entre elle provient du schéma d'intégration : connaissant la valeur exacte du potentiel newtonien et de ses dérivées spatiales à toute position :

$$\Phi(r) = -\frac{GM}{r} \qquad \text{et} \qquad \frac{\mathrm{d}\Phi(r)}{\mathrm{d}r} = \frac{GM}{r^2} \tag{7.10}$$

il est possible d'intégrer numériquement les géodésiques nulles en coordonnées cartésiennes en utilisant le raytracer sans la grille. C'est-à-dire que pour chaque pas d'intégration, au lieu de calculer le potentiel et ses dérivées par interpolation des quantités discrétisées sur la grille AMR, les valeurs exactes sont injectées dans l'intégrateur. Ce faisant, on peut donc accéder aux erreurs numériques liées à l'intégrateur seul. Une fois celles-ci évaluées, ce sont les erreurs liées à l'introduction de la grille dans le problème qui peuvent être estimées et notamment les aspects de discrétisation et d'interpolation. Pour le premier, il s'agit de s'intéresser à la façon dont est échantillonné le potentiel autour de la source suivant les critères de raffinement choisis et les conséquences que cela a sur le calcul de la trajectoire. On peut facilement concevoir que lorsque le paramètre d'impact est de l'ordre de la taille d'une ou deux cellules coarse, le raffinement adaptatif va jouer un rôle essentiel sur le calcul de $\delta(r)$, en particulier à l'approche de r_0. Pour le second, il s'agit essentiellement de s'interroger sur l'influence du schéma d'interpolation choisi : NGP ou CIC. Pour certains cas pathologiques d'un problème qui expose des symétries comme la métrique de Schwarzschild, le meilleur schéma d'interpolation n'est en effet pas toujours celui que l'on croit.

Pour mesurer ces différentes sources d'erreur la méthode adoptée est la suivante : on se place dans une boîte de simulation cubique de côté L, centrée en $(x_s, y_s, z_s) = (0,0,0)$ sur la masse de Schwarzschild, et assez grande pour sonder des $\delta(r)$ pour r grand par rapport à r_0. Au sein de cette

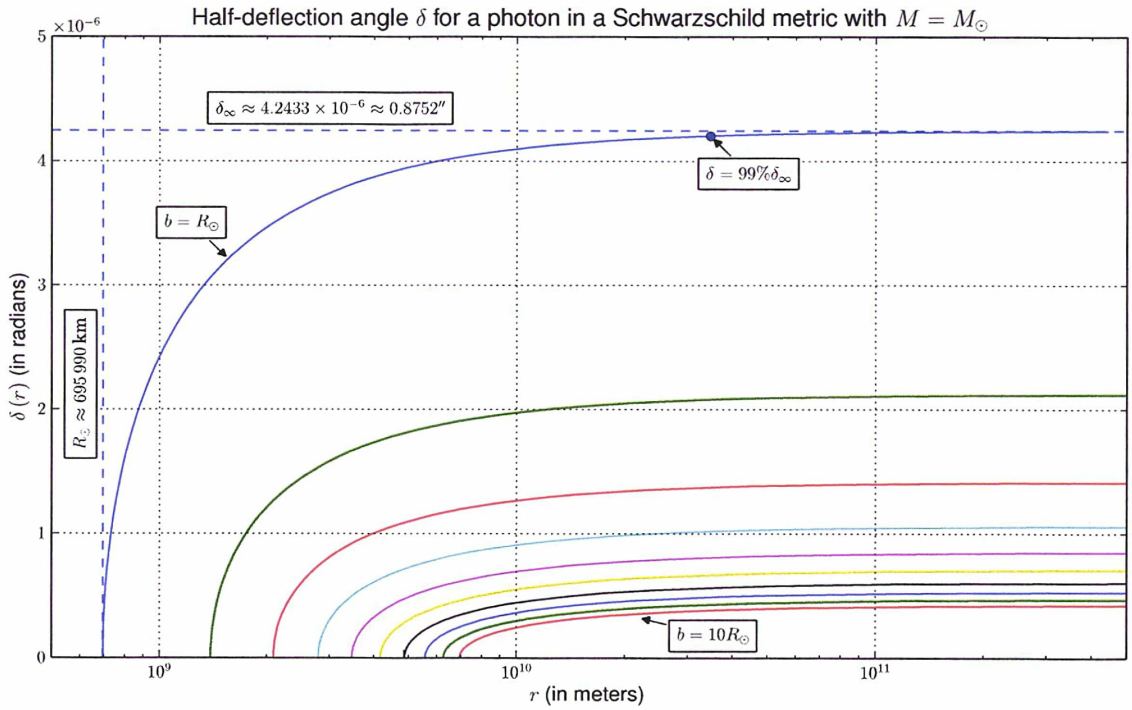


FIGURE 75 – Angle de déflexion dans le cas d'une métrique de Schwarzschild associée au Soleil. Les dix courbes correspondent aux $\delta(r)$ associés à dix paramètres d'impacts allant de $b = R_\odot$ à $b = 10R_\odot$, calculés en utilisant les fonctions de Weierstrass et les méthodes numériques décrites précédemment. Pour le cas $b = R_\odot$, 99% de la valeur finale est atteinte pour $r \approx 50b$: c'est-à-dire que dans le cas d'une boîte de simulation centrée sur le Soleil de 100 fois la taille du paramètre d'impact, seule une précision relative de 10^{-2} sur l'angle de déflexion total pourrait être atteinte. Pour un halo de masse $M_{200} = 10^{14} M_\odot$ et pour $b = r_{200}$, $\delta(r)$ a le même profil et tend vers $\delta_\infty \approx 1.0185 \times 10^{-5} \approx 2.1''$.

boîte, on lance un photon avec les conditions initiales :

$$\begin{cases} x_0 = x_s - \frac{L}{2} = -\frac{L}{2} \\ y_0 = y_s + \tilde{b} = \tilde{b} \\ z_0 = z_s = 0 \end{cases} \quad \text{et} \quad \begin{cases} \left(\frac{dx}{d\lambda}\right)_0 = \kappa \\ \left(\frac{dy}{d\lambda}\right)_0 = 0 \\ \left(\frac{dz}{d\lambda}\right)_0 = 0 \end{cases} \tag{7.11}$$

où $\tilde{b}$ représente le paramètre d'impact approché, et où κ est normalisé de façon à ce que l'équation des géodésiques soit initialement satisfaite par le photon. Etant donné que le photon n'est pas lancé depuis une distance infinie de la masse, seule une approximation de b peut être donnée, d'où la notation $\tilde{b}$. Le photon va donc se propager jusqu'à la masse centrale, être dévié puis s'en éloigner jusqu'à la face opposée de la boîte. La question qui se pose naturellement est donc celle de la mesure de $\delta(r)$ en étant le moins dépendant possible des conditions initiales, et notamment de $\tilde{b}$. Et la solution est simple : il va s'agir d'utiliser r_0 et non $\tilde{b}$. En effet, le long de la géodésique nulle calculée numériquement, il est possible de déterminer la distance d'approche maximale r_0 et donc de déterminer le paramètre d'impact « réel » via la relation 7.4. On peut ensuite resymétriser le problème par rapport à r_0 sur le principe de la figure 73 : pour un r donné, on peut évaluer les deux positions correspondantes sur la trajectoire en coordonnées cartésiennes :

$$\begin{cases} \vec{r}_1 = (x_1 - x_s)\,\vec{e}_x + (y_1 - y_s)\,\vec{e}_y + (z_1 - z_s)\,\vec{e}_z \\ \vec{r}_2 = (x_2 - x_s)\,\vec{e}_x + (y_2 - y_s)\,\vec{e}_y + (z_2 - z_s)\,\vec{e}_z \end{cases} \quad \text{avec} \quad \|\vec{r}_1\| = \|\vec{r}_2\| = r \tag{7.12}$$

et donc calculer, avec :

$$\boxed{2\phi\left(r\right) = \tan^{-1}\left(\frac{\|\vec{r}_1 \times \vec{r}_2\|}{\vec{r}_1 \cdot \vec{r}_2}\right) \quad \Rightarrow \quad \delta\left(r\right) = \frac{1}{2}\tan^{-1}\left(\frac{\|\vec{r}_1 \times \vec{r}_2\|}{\vec{r}_1 \cdot \vec{r}_2}\right) - \cos^{-1}\left(\frac{r_0}{r}\right)} \tag{7.13}$$

où numériquement $\tan^{-1}\left(\frac{y}{x}\right)$ correspond à `atan2(y, x)`. Si l'on appelle cette quantité δ_{num} et δ_{th} celle donnée en 7.9, il ne reste ensuite plus qu'à évaluer l'erreur relative sur l'angle de déflexion :

$$\epsilon_\delta = \left|\frac{\delta_{\text{num}} - \delta_{\text{th}}}{\delta_{\text{th}}}\right| \tag{7.14}$$

En plus de cela, deux autres erreurs peuvent être évaluées le long de chaque trajectoire : l'erreur relative ϵ_Φ entre le potentiel 7.10 et le potentiel réinterpolé à partir de la grille d'une part, et l'erreur relative $\epsilon_{\mathrm{d}s^2}$ sur $\mathrm{d}s^2$ dont la valeur doit être nulle le long de toute la trajectoire d'autre part. Pour cette dernière, plutôt que de calculer directement $\mathrm{d}s^2$, il s'agit de mesurer le rapport entre les parties spatiales et temporelles de la métrique faiblement perturbée :

$$\epsilon_{\mathrm{d}s^2} = \left|\frac{\left(1 - 2\frac{\Phi}{c^2}\right)\left(\mathrm{d}x^2 + \mathrm{d}y^2 + \mathrm{d}z^2\right)}{c^2\left(1 + 2\frac{\Phi}{c^2}\right)\mathrm{d}t^2} - 1\right| \tag{7.15}$$

Toute la méthode présentée ici est générique en ce sens qu'elle fournit tous les outils, tant analytiques que numériques pour évaluer les erreurs en métrique de Schwarzschild. Elle pourrait donc donner lieu à une analyse fouillée et détaillée de toutes les sources d'erreur au sein de l'algorithme. Une telle analyse dépasse le cadre de cette thèse, car il s'agissait ici simplement de valider l'intégrateur. Elle se présente cependant comme une extension naturelle de cette dernière par application de la méthode décrite précédemment. La vérification désirée, elle, a été obtenue par raytracing en présence d'un halo de masse $M = 10^{14} M_\odot$ pour différents paramètres de grille et pour différents paramètres d'impact. En régime faiblement perturbé, et pour une grille raffinée selon les mêmes critères que la grille du Full Universe Run, des erreurs strictement inférieures à 10^{-12} sur les angles de déflexion ont été mesurées dans tous les cas.

7.3 Application cosmologique

7.3.1 Intégration des géodésiques dans les cônes de lumière du Full Universe Run

Un code d'intégration des géodésiques en métrique de champ faible basé sur MAGRATHEA et sur une architecture logicielle à deux niveaux, statique et cinématique ayant été développé et testé conformément aux paragraphes précédent, il devient possible de le mettre en application dans le cadre du Full Universe Run. Il ne s'agit pas ici de rendre compte de tous les aspects de cette mise en application, mais plutôt de mettre en avant quelques éléments importants de l'utilisation du code qui a été faite.

Le code a été utilisé sur les nœuds fins de CURIE, là où a été réalisé le Full Universe Run, et par conséquent, là où sont disponible les données brutes. En tout, six cônes de lumières, deux par cosmologie, ont été analysés. Chacun d'entre eux représente $16^3 = 4096$ fichiers cubes issus de RAMSES pour un poids total allant de 6 Tio à 10 Tio. Avec un tel volume de données à traiter, se pose la question du nombre de tâches minimal nécessaire pour charger la simulation en mémoire, construire l'arbre correspondant, et effectuer le raytracing. Il apparait que 4096 tâches à 4 Gio par tâche est une configuration trop limitée. Par conséquent, c'est plutôt une configuration à 8192 tâches et threads qui a été choisie. Chaque nœud de calcul sur la machine CURIE est composé de 2 processeurs de 8 cœurs chacun et embarque un total de 64 Gio de mémoire. Les premiers tests avaient été effectués pour 512 tâches MPI et 16 threads par tâche. Cependant, pour des raisons de taille d'arbre local, de volume de données à

lire pour maximiser l'utilisation de la bande passante, et d'optimisation vis-à-vis du NUMA [5] au sein de chaque nœud il s'est avéré être plus performant d'utiliser 1024 tâches MPI et 8 threads par tâche, ou, autrement dit, une tâche MPI par processeur et un thread par cœur, l'Hyperthreading étant désactivé sur la machine.

L'étape de lecture des fichiers et de redistribution des cellules sur les bonnes tâches en fonction des zones spatiales coniques s'est révélée particulièrement coûteuse, avec un temps d'exécution pouvant aller jusqu'à plusieurs heures. Il apparait que le tri, la répartition des cellules, et la construction de l'arbre ne sont pas en cause : c'est la lecture des données et les accès simultanés à certains fichiers qui semblent être à l'origine du problème. Même si dans notre cas, ce temps d'exécution n'a pas empêché les analyses, une étude des performances plus détaillée serait nécessaire. Aussi, la sauvegarde de la structure de l'arbre dans des fichiers temporaires une fois la redistribution terminée a été particulièrement utile, le temps de démarrage passant de plus d'une heure à trois ou quatre minutes dans le cas où un chargement depuis ces fichiers temporaires était possible.

Au niveau de l'intégration, la plupart des analyses ont été effectuées avec 1 000 000 de bundles de 8 photons autour du photon central, soit 9 000 000 de trajectoires, avec un temps typique d'exécution de l'ordre de la demi-heure, chargement des données depuis des fichiers temporaires compris. Deux simulations à 10 000 000 de bundles ont permis de s'assurer de la robustesse des résultats, pour un temps d'exécution typique de l'ordre de deux heures. Cela conduit à temps moyen d'intégration d'environ 700 ms par trajectoire, dominé par le calcul des interpolations par Cloud-In-Cell inverse, ces derniers nécessitant de nombreuses recherches de voisins et des décryptages d'indices répétés. La mise à disposition des jeux d'instructions de manipulation de bits sur les prochaines générations de supercalculateurs devrait permettre de diminuer le coût de ces opérations relativement au temps d'intégration total.

Enfin, originellement, et lors des premiers tests, l'ensemble des trajectoires était enregistré sur disque, la phase d'analyse statistique étant prise en charge par des routines PYTHON. Au-delà de 10 000 trajectoires cette méthode a vite montré ces limites, le temps de lecture des données devenant prohibitif. C'est pourquoi, pour économiser une étape d'écriture et de relecture inutile, et pour tirer parti de la puissance de calcul disponible sur CURIE, la réduction des données et le calcul de la moyenne et de l'écart-type de la distance de diamètre angulaire en fonction du redshift a été implémenté directement dans le code, faisant passer le temps d'analyse de plusieurs heures à quelques secondes.

7.3.2 Perspectives

Le code étant pleinement fonctionnel, et sa bonne scalabilité ayant été testée avec succès jusqu'à 8192 tâches et threads, de nombreuses applications physiques sont possibles. Certaines d'entre elles ont déjà été décrites au quatrième chapitre. L'analyse des données du troisième observateur de chaque modèle cosmologique du Full Universe Run est à prévoir, de même que l'analyse des « samples » extraits à chaque pas de temps qui permettront d'avoir directement accès à la variation du potentiel gravitationnel dans le temps. À moyen terme, des analyses plus fournies de weak-lensing devraient être possibles en utilisant les informations sur les déformations progressives des bundles avec le redshift.

D'un point de vue numérique, ce code a permis de démontrer la viabilité des approches décrites dans les chapitres précédents. Tout d'abord, il a permis de vérifier la pertinence de la métaprogrammation par template pour maximiser la généricité et les performances des codes dans un contexte de calcul haute performance. Ensuite, il a conduit à tester le bon fonctionnement de MAGRATHEA malgré le fait que celle-ci soit une bibliothèque « exploratoire » dans le sens où elle a avant tout été utilisée

5. Non Uniform Memory Access.

pour tester un certain nombre de techniques d'implémentation non triviales. Enfin, ce code a permis de tester la pertinence d'une architecture logicielle en trois niveaux de complexité, le premier ayant conduit à l'implémentation d'un nouvel algorithme de gestion du raffinement adaptatif et le second consistant ici en les routines d'intégration génériques. L'algorithme d'orthtree s'est avéré être pleinement fonctionnel dans le contexte applicatif du raytracing en métrique de champ faible. Enfin, les investigations sur l'implémentation de fonctions liées aux aspects cinématiques devraient être poursuivies pour former un ensemble cohérent susceptible d'être pleinement intégré à MAGRATHEA, avant de passer, à plus long terme, aux aspects dynamiques.

Chapitre 8

Conclusion

***

La compréhension de l'origine de l'accélération de l'expansion de l'Univers est l'une des grandes questions de la cosmologie contemporaine. Pour y répondre, l'analyse de la structuration à grande échelle a un rôle clef à jouer car cette dernière garde l'empreinte des mécanismes physiques sous-jacents qui ont conduit à cette accélération. Comprendre cette empreinte, et comprendre comment elle est encodée dans les observations, c'est être à même de déterminer les observables pertinentes pour sonder les propriétés et la nature de ces mécanismes physiques. C'est dans ce contexte que cette thèse s'est interrogée sur la propagation de la lumière aux grandes échelles, et sur l'empreinte laissée par l'accélération de l'expansion dans les observations.

Un accent particulier a été mis sur la méthodologie en vue de garantir un maximum de robustesse et de généricité à la démarche dans un contexte théorique foisonnant. D'un point de vue physique, nous nous sommes attaché à rester au plus près des hypothèses de base de manière à éviter un maximum d'approximations et de manière à pouvoir effectuer des analyses d'effets très divers au sein d'un cadre théorique commun. Ainsi, nous avons travaillé directement sur les équations géodésiques en champ faible afin de traiter la propagation de la lumière par une approche de raytracing relativiste compatible avec les simulations cosmologiques. Une approche *ab initio*, en quelque sorte, en ce sens qu'il s'agit de retrouver une très grande variété de phénomènes observables comme autant de propriétés émergentes de principes fondamentaux : en l'occurrence l'application des équations de la relativité générale à une distribution de potentiel gravitationnel issue de simulations telles que le Full Universe Run. Ce dernier a permis, pour la première fois, de sonder la structuration de la matière à l'échelle de l'Univers Observable, et ce, pour trois modèles cosmologiques différents. Mais il a permis bien plus que ça : il a permis de réinterroger la cosmologie numérique et ses méthodes. Car bâtir des résultats numériques solides dans le cadre d'une approche de théorie assistée par ordinateur ne peut faire l'économie de codes solides et robustes dans un contexte où la complexité de ces derniers ne cesse de croître.

Pour ces raisons, nous avons proposé et exploré l'approche de compilateurs et de langage dédiés à la cosmologie sous la forme de langages embarqués et de bibliothèques actives. Techniquement, cela a consisté à approndir la piste de la métaprogrammation par templates en C++ pour finalement conduire à la création d'une nouvelle bibliothèque d'analyse, baptisée Magrathea, et dédiée à la mise en place de ces concepts dans le cadre applicatif de la cosmologie. Cela a permis de valider la viabilité de ces techniques pour atteindre un maximum de généricité et garantir un maximum de performances sur les supercalculateurs. Les leçons du Full Universe Run et les réflexions sur les bibliothèques actives nous ont également conduit à proposer une nouvelle architecture logicielle pour la cosmologie numérique en trois niveaux de complexité : statique, cinématique et dynamique. Le suivi des photons dans un Univers structuré s'est inscrit dans le cadre des deux premiers niveaux mais les développements mis en œuvre peuvent être vus comme faisant parti d'une démarche plus ambitieuse visant à proposer, à terme, les bases d'un code dynamique générique et modulaire adapté aux futures architectures exas-

cale. Cela apparait tout du moins comme une extension naturelle de ce travail de thèse.

Au niveau de la partie statique, un nouvel algorithme de gestion des arbres adaptatifs basé sur une dualité programme/nombre entier a été conçu et implémenté avec des gains en termes de performances et d'occupation mémoire significatifs. L'algorithme a été utilisé pour traiter les cônes de lumière du Full Universe Run, soit plusieurs centaines de milliards de cellules sur près de 10 000 processeurs. Les perspectives données par cet algorithme sont nombreuses pour toutes les analyses et simulations nécessitant un maillage hypercubique. Il ouvre également des perspectives quant à la création d'une bibliothèque unifiée pour tous les types d'arbres basée sur le même genre d'approches. Pour la partie cinématique, les techniques de métaprogrammation par template ont été utilisées pour concevoir un code de raytracing 3D en champ faible. Ce code a été utilisé sur le supercalculateur Curie pour analyser la propagation de la lumière dans le champ gravitationnel du Full Universe Run.

Ces développements analytiques et numériques nous ont finalement permis de suivre les trajectoires de photons autour d'observateurs virtuels issus de trois modèles cosmologiques différents : le modèle de concordance ΛCDM, un modèle de quintessence RPCDM et un modèle d'énergie noire phantom WCDM. Parmi les multiples observables qui auraient pu faire l'objet d'analyses dès lors que l'on dispose d'un code qui intègre directement les équations géodésiques, nous nous sommes concentrés sur la mesure des distances dans l'Univers. Ce choix est motivé par le fait que les distances de diamètre angulaires et les distances de luminosité constituent des observables centrales dans la détermination de l'évolution du paramètre d'échelle. Or ces dernières sont systématiquement évaluées dans un cadre purement friedmannien, ce que l'Univers n'est rigoureusement pas. En effet, la trajectoire de la lumière est perturbée par rapport au cas homogène et isotrope. Cette perturbation étant liée à la structuration, donc au modèle cosmologique, la propagation de la lumière garde l'empreinte de ce dernier. C'est du moins ce qu'ont permis de confirmer les analyses du Full Universe Run. Aussi, deux types d'effets ont pu être évalués : d'une part, l'effet des inhomogénéités sur la moyenne des distances de diamètre angulaire, et d'autre part, l'effet des inhomogénéités sur l'écart-type de ces mêmes distances. Si le premier est mesurable au sein des simulations, il ne constitue toutefois pas une observable au sens observationnel. Il permet en revanche de constater que les inhomogénéités ont tendance à rehausser les valeurs des distances de diamètre angulaire si bien qu'un modèle peu structuré comme RPCDM pour lequel les inhomogénéités ont été prises en compte peut être interprété comme un modèle de concordance friedmannien. L'écart-type, ou plus généralement, la distribution des distances de diamètre angulaire, est, elle aussi, affectée par les inhomogénéités. Les mesures effectuées dans le Full Universe Run indiquent des différences de l'ordre de 3% à $z = 2$ pour ΛCDM entre le cas inhomogène et homogène, tandis qu'elles sont plutôt de l'ordre de 4% pour WCDM et 2% pour RPCDM. Or le spectre de puissance des distances de diamètre angulaire et des distances de luminosité est, lui, bien observable. Sa mesure serait donc à même de venir apposer des contraintes supplémentaires sur les modèles cosmologiques et sur la nature de la physique entrainant l'accélération de l'expansion. Cela nécessiterait d'étendre les analyses physiques menées ici pour quantifier l'influence de la prise en compte des inhomogénéités sur la mesure de chacun des paramètres cosmologiques.

Si pour la première fois, l'intégration des géodésiques en champ faible permet d'estimer numériquement l'amplitude des effets des inhomogénéités sur les distances cosmologiques sur des échelles de l'ordre de la taille de l'Univers Observable, elle ouvre également la porte à de nombreuses perspectives. L'une des questions majeures qui a été soulevée dans ce manuscrit a trait à la définition rigoureuse des distances cosmologiques hors du cadre homogène. En effet, d'un point de vue observationnel, il semble exister de multiples critères d'arrêt pour des faisceaux de photons. Dès lors, la non-équivalence de ces définitions entraîne des biais différents en fonction des observables étudiées. Il conviendrait donc de s'interroger sur la prise en compte de ces différences dans le cadre de l'interprétation des observations. Outre de telles questions, le travail présenté ici conduit à de nombreuses possibilités d'analyses. Ne serait-ce que le lentillage gravitationnel faible qui sera au cœur de la mission spatiale Euclid. Ou

l'étude de l'effet Sachs-Wolfe intégré. Ou même encore les analyses de redshift drift telles qu'elles sont à l'étude dans le contexte de l'European Extremely Large Telescope. Il s'agit là de quelques pistes permises par la généricité de la méthodologie mise en œuvre, tant au niveau physique qu'au niveau numérique, qui dépassent largement le cadre de cette thèse, mais qui s'inscrivent néanmoins en ligne direct du travail présenté ici.

Enfin, pour clore ce manuscrit, si l'on demandait à l'auteur de ces pages ce qui lui tient le plus à cœur, il répondrait sans doute « explorer la piste d'une cosmologie *ab initio* ». La cosmologie du vingtième siècle s'est construite comme une science intensive au sens d'Anderson[144] dans le sens où l'on cherchait à mettre en évidence des symétries fondamentales pour construire une représentation cohérente de notre Univers et de son évolution. Mais notre Univers n'est pas symétrique : il est à la fois complexe et changeant. Le numérique ne doit pas être le prétexte d'une science centrée sur des « recettes de cuisine » : il doit être au contraire l'occasion de déconstruire méthodiquement les symétries des lois physiques fondamentales dans le cadre de systèmes complexes. Et c'est à ce titre qu'il semble à même de fournir les outils d'un traitement extensif de la cosmologie. Aussi, à l'heure du petascale et de l'exascale, le numérique ouvre les portes d'une cosmologie reconstruite comme une propriété émergente de la relativité générale prise en compte dans toute sa généralité et sa complexité. La route d'un code de simulation cosmologique complètement relativiste semble encore longue et tortueuse. Et c'est justement pour cette raison qu'elle vaut la peine d'être explorée. Car ce sont toujours les chemins longs et tortueux qui réservent le plus de surprises.

Annexe A

Développements autour de la gravité de Brans-Dicke

***

A.1 Théorie de Brans-Dicke et hypothèse AWE . . . 179
A.2 Expression des actions . . . 179
A.3 Application à la cosmologie . . . 180
A.4 Solutions analytiques . . . 181
A.5 Solutions numériques . . . 183
A.6 Résultats . . . 185
A.7 Conclusion . . . 187

A.1 Théorie de Brans-Dicke et hypothèse AWE

Dans la suite, on se penche sur l'hypothèse AWE[42], une généralisation des théories scalaire-tenseurs envisageant des couplages non universels. Dans ce cadre, le contenu de l'Univers est divisé en trois secteurs différents : la gravitation, décrite par la théorie de Brans-Dicke, le secteur visible, représenté principalement par les baryons et les photons, et le secteur invisible constitué par « une énergie anormalement pesante » (Abnormally Weighting Energy). Une conséquence de ce modèle est de fournir une description unifiée de la matière noire et de l'énergie noire par une relaxation du principe d'équivalence faible du secteur invisible aux grandes échelles. L'effet de l'espace-temps d'arrière-plan est alors différent sur le secteur invisible et le secteur visible, ou dit autrement, les couplages de la gravitation avec le secteur invisible et le secteur visible sont différents. L'avantage de cette modélisation est qu'elle conduit à l'obtention de comportements accélérés dans l'Univers tardif ainsi que dans l'Univers primordial dans le contexte inflationnaire[42]. Toutefois, la dérivation de solutions explicites n'est pas des plus triviales. Aussi, ce qui suit est le résultat d'une collaboration avec Jean-Michel Alimi et Anastasia Golubtsova dans le cadre de laquelle j'ai contribué à l'obtention de solutions numériques. L'accent est donc mis sur cet aspect, les détails de la dérivation des expressions analytiques résumée ici pouvant être trouvés dans l'article associé[43].

A.2 Expression des actions

Pour explorer ce type de comportements, nous allons considérer l'action de la théorie de Brans-Dicke généralisée[40, 42]. Dans le formalisme des théories scalaire-tenseurs celle-ci peux s'exprimer dans

deux représentations différentes : celle de Jordan, notée dans la suite JF pour « Jordan Frame » et celle d'Einstein, notée dans la suite EF pour « Einstein Frame ». La seconde peut se déduire de la première par une transformation conforme de la forme $g_{\mu\nu}^{JF} = F\left(\varphi^{EF}\right) g_{\mu\nu}^{EF}$. Dans le contexte dans lequel on se place, les observables doivent être calculés dans le référentiel de Jordan, car il est le seul à avoir une signification physique. Le problème est qu'il intrique les degrés de liberté scalaire et tensoriel, si bien que le référentiel d'Einstein est bien plus commode du point de vue des calculs. Aussi, pour déterminer l'évolution du facteur d'échelle dans la représentation de Jordan, nous calculerons d'abord son évolution en représentation d'Einstein avant d'y appliquer les transformations adéquates. Dans la représentation de Jordan, nous partons ainsi d'une action de la forme suivante :

$$\begin{aligned} S_{JF} =& \frac{c^3}{16\pi\tilde{G}} \int \mathrm{d}^4x\sqrt{-\tilde{g}} \left(\Phi\tilde{R} - \frac{\omega_{BD}(M(\Phi))}{\Phi}\tilde{g}^{\mu\nu}\frac{\partial\Phi}{\partial x^\mu}\frac{\partial\Phi}{\partial x^\nu}\right) \\ &+ \varepsilon_1 S_m\left[\psi_m, \tilde{g}_{\mu\nu}\right] + \varepsilon_2 S_a\left[\psi_a, M(\Phi)^2\tilde{g}_{\mu\nu}\right] \end{aligned} \tag{A.1}$$

où :

- $\tilde{G}$ est la constante de gravitation « nue », qui diffère de la constante de gravitation effective G_{eff}
- $\tilde{g}_{\mu\nu}$ est la métrique du référentiel de Jordan couplée au champ de matière ordinaire
- $\tilde{g}$ est son déterminant
- $\tilde{R}$ est le scalaire de courbure qui lui est associé
- Φ est le degré de liberté scalaire tel que la constante de gravitation observée vaut $G_{\text{eff}} = \frac{\tilde{G}}{\Phi}$
- ω_{BD} est la fonction de couplage de Brans-Dicke
- S_m est l'action de la matière, reposant sur le champ ψ_m
- S_a est celle du secteur AWE, reposant sur le champ ψ_a
- ε_1 et ε_2 représentent les signes des termes cinétiques des champs scalaires : $\varepsilon = +1$ modélise un champ classique, $\varepsilon = -1$ un champ « phantom »

Pour réexprimer l'action dans la représentation d'Einstein, on considère la transformation conforme :

$$\tilde{g}_{\mu\nu} = A_m^2(\varphi) g_{\mu\nu} \qquad \text{et} \qquad \begin{cases} \Phi = \frac{\tilde{G}}{G} = A_m^{-2}(\varphi) \\ (3+2\omega_{BD})^2 = \frac{\mathrm{d}\ln A_m(\varphi)}{\mathrm{d}\varphi} \\ M(\Phi) = \frac{A_a(\varphi)}{A_m(\varphi)} \end{cases} \tag{A.2}$$

si bien que l'action peut se réécrire dans la représentation d'Einstein :

$$\begin{aligned} S_{EF} =& \frac{c^3}{16\pi G} \int d^4x\sqrt{-g} \left(R - 2g^{\mu\nu}\frac{\partial\varphi}{\partial x^\mu}\frac{\partial\varphi}{\partial x^\nu}\right) \\ &- \int d^4x\sqrt{-g}\varepsilon_1 A_m^2(\varphi)\, g^{\mu\nu}\frac{\partial\psi_m}{\partial x^\mu}\frac{\partial\psi_m}{\partial x^\nu} \\ &- \int d^4x\sqrt{-g}\varepsilon_2 A_a^2(\varphi)\, g^{\mu\nu}\frac{\partial\psi_a}{\partial x^\mu}\frac{\partial\psi_a}{\partial x^\nu} \end{aligned} \tag{A.3}$$

où $A_m(\varphi) > 0$ et $A_a(\varphi) > 0$ sont les fonctions de couplage non-minimal pour le secteur visible et invisible.

A.3 Application à la cosmologie

Afin de mettre en lumière la dynamique d'un tel modèle dans le cadre cosmologique, on considère une métrique FLRW de courbure nulle :

$$\mathrm{d}s^2 = g_{\mu\nu}\,\mathrm{d}x^\mu\,\mathrm{d}x^\nu = -c^2\,\mathrm{d}t^2 + a^2(t)\delta_{ij}\,\mathrm{d}x^i\,\mathrm{d}x^j \tag{A.4}$$

et l'on suppose que les champs scalaires dépendent uniquement d'une coordonnée temporelle. Dans ces conditions [1], le facteur d'échelle dans le référentiel d'Einstein peut se mettre sous la forme :

$$a_{EF}(t) = a_0 \sqrt[3]{3H_0(t - t_0) + 1} \tag{A.5}$$

où a_0, H_0 et t_0 sont des constantes d'intégration. Dans ces conditions toujours, le champ scalaire φ vérifie :

$$\int_{\varphi_0}^{\varphi} \frac{\mathrm{d}\phi}{\sqrt{2E_\phi - V(\phi)}} = \tau \qquad \text{avec} \qquad \begin{cases} \tau = \ln\left(\frac{t}{t_0}\right) \\ E_\varphi = \frac{1}{2}\left(\left(\frac{\mathrm{d}\varphi}{\mathrm{d}\tau}\right)^2 + V(\varphi)\right) \\ V(\varphi) = \varepsilon_1 \frac{C_m^2}{A_m^2(\varphi)} + \varepsilon_2 \frac{C_a^2}{A_a^2(\varphi)} \end{cases} \tag{A.6}$$

où C_m et C_a sont des constantes du mouvement :

$$C_m = A_m(\varphi) \frac{\sqrt{8\pi G}}{c^{3/2}} \frac{\mathrm{d}\psi_m}{\mathrm{d}\tau} \qquad \text{et} \qquad C_a = A_a(\varphi) \frac{\sqrt{8\pi G}}{c^{3/2}} \frac{\mathrm{d}\psi_a}{\mathrm{d}\tau} \tag{A.7}$$

A.4 Solutions analytiques

Dans la suite, on considère la forme suivante pour les fonctions de couplage :

$$\boxed{A_m(\varphi) = \frac{1}{A_a(\varphi)} = e^{k_m \varphi}} \tag{A.8}$$

avec k_m la constante de couplage. Cette forme est inspirée des modèles de supergravité et est motivée par le fait que des comportements accélérés dans le référentiel de Jordan sont observés lorsque A_m et A_a sont liées par une relation de proportionnalité inverse[42]. Dès lors, $V(\varphi)$ peut se réécrire sous la forme :

$$V(\varphi) = \varepsilon_1 C_m^2 e^{-2k_m\varphi} + \varepsilon_2 C_a^2 e^{2k_m\varphi} \tag{A.9}$$

si bien que l'équation A.6 donne, par changement de variables :

$$\int_{z_0}^{z} \frac{\mathrm{d}x}{\sqrt{ax^4 + bx^2 + c}} = \tau \qquad \text{avec} \qquad \begin{cases} z = e^{k_m\varphi} \\ a = -\varepsilon_2 k_m^2 C_a^2 \\ b = 2k_m^2 E_\varphi \\ c = -\varepsilon_1 k_m^2 C_m^2 \end{cases} \tag{A.10}$$

Cinq cas peuvent alors être distingués analytiquement suivant les signes de ε_1 et ε_2 et le domaine de définition de l'énergie E_φ. Les solutions exactes pour φ et $A_m(\varphi)$ sont résumées en figure 76. Elles s'expriment à partir des fonctions elliptiques de Jacobi sn et cn elles-mêmes définies à partir des intégrales elliptiques incomplètes du premier type :

$$\begin{aligned} \mathrm{sn}(u,k) &= \sin(\psi) \\ \mathrm{cn}(u,k) &= \cos(\psi) \end{aligned} \qquad \text{pour} \qquad u = \int_0^{\psi} \frac{\mathrm{d}\theta}{\sqrt{1 - k^2 \sin(\theta)}} \qquad \text{avec} \qquad 0 \leq k < 1 \tag{A.11}$$

Mais pourquoi avoir fait tout ce travail alors que l'évolution du facteur d'échelle est connue et s'exprime très simplement selon l'équation A.5 ? Le problème est que l'évolution du facteur d'échelle dans le référentiel d'Einstein ne dit rien de son évolution dans le référentiel physique de Jordan. Ou

1. Les détails des calculs sont donnés dans J.-M. Alimi, A.A. Golubtsova et V. Reverdy 2013[43].

Brans-Dicke gravity solution : case 1 : $\varepsilon_1 = +1, \varepsilon_2 = +1, E_\varphi \in \mathbb{R}_+$	
a,b,c	$a \in \mathbb{R}_-^*, \quad b \in \mathbb{R}_+^*, \quad c \in \mathbb{R}_-^*, \quad b^2 > 4ac$
ρ,λ	$\rho^2 = \frac{b-\sqrt{b^2-4ac}}{2\lvert a\rvert}, \quad \lambda^2 = \frac{b+\sqrt{b^2-4ac}}{2\lvert a\rvert}, \quad \rho^2 < \lambda^2, \quad 0 < \rho \leq A_m \leq \lambda$
φ,k	$\varphi = \frac{1}{k_m} \ln\left(\frac{\rho\lambda \mathrm{sn}\left(\sqrt{\lvert a\rvert(\lambda^2+\rho^2)}\tau,k\right)}{\sqrt{\lambda^2+\rho^2-\rho^2\mathrm{sn}^2\left(\sqrt{\lvert a\rvert(\lambda^2+\rho^2)}\tau,k\right)}}\right), \quad k = \frac{\sqrt{\lambda^2-\rho^2}}{\lambda}$
A_m,A_a	$A_m(\varphi) = \frac{1}{A_a(\varphi)} = \frac{\rho\lambda \mathrm{sn}\left(\sqrt{\lvert a\rvert(\lambda^2+\rho^2)}\tau,k\right)}{\sqrt{\lambda^2+\rho^2-\rho^2\mathrm{sn}^2\left(\sqrt{\lvert a\rvert(\lambda^2+\rho^2)}\tau,k\right)}}$

Brans-Dicke gravity solution : case 2 : $\varepsilon_1 = -1, \varepsilon_2 = +1, E_\varphi \in \mathbb{R}$	
a,b,c	$a \in \mathbb{R}_-^*, \quad b \in \mathbb{R}, \quad c \in \mathbb{R}_+^*$
ρ,λ	$\rho^2 = \frac{b+\sqrt{b^2-4ac}}{2\lvert a\rvert}, \quad \lambda^2 = \frac{-b+\sqrt{b^2-4ac}}{2\lvert a\rvert}, \quad 0 < A_m \leq \rho$
φ,k	$\varphi = \frac{1}{k_m} \ln\left(\frac{\lambda\rho}{\sqrt{\lambda^2-\lambda^2\mathrm{sn}^2\left(\sqrt{\lvert a\rvert}\lambda\tau,k\right)+\rho^2\mathrm{sn}^2\left(\sqrt{\lvert a\rvert}\lambda\tau,k\right)}}\right), \quad k = \frac{\rho}{\sqrt{\lambda^2+\rho^2}}$
A_m,A_a	$A_m(\varphi) = \frac{1}{A_a(\varphi)} = \frac{\lambda\rho}{\sqrt{\lambda^2-\lambda^2\mathrm{sn}^2\left(\sqrt{\lvert a\rvert}\lambda\tau,k\right)+\rho^2\mathrm{sn}^2\left(\sqrt{\lvert a\rvert}\lambda\tau,k\right)}}$

Brans-Dicke gravity solution : case 3 : $\varepsilon_1 = +1, \varepsilon_2 = -1, E_\varphi \in \mathbb{R}$	
a,b,c	$a \in \mathbb{R}_+^*, \quad b \in \mathbb{R}, \quad c \in \mathbb{R}_-^*$
ρ,λ	$\rho^2 = \frac{b+\sqrt{b^2-4ac}}{2a}, \quad \lambda^2 = \frac{-b+\sqrt{b^2-4ac}}{2a}, \quad 0 < \lambda \leq A_m$
φ,k	$\varphi = \frac{1}{k_m} \ln\left(\frac{\lambda}{\mathrm{cn}\left(\sqrt{a(\lambda^2+\rho^2)}\tau,k\right)}\right), \quad k = \frac{\rho}{\sqrt{\lambda^2+\rho^2}}$
A_m,A_a	$A_m(\varphi) = \frac{1}{A_a(\varphi)} = \frac{\lambda}{\mathrm{cn}\left(\sqrt{a(\lambda^2+\rho^2)}\tau,k\right)}$

Brans-Dicke gravity solution : case 4 : $\varepsilon_1 = -1, \varepsilon_2 = -1, E_\varphi \in \mathbb{R}_+^*$	
a,b,c	$a \in \mathbb{R}_+^*, \quad b \in \mathbb{R}_+^*, \quad c \in \mathbb{R}_+^*, \quad b^2 > 4ac$
ρ,λ	$\rho^2 = \frac{b+\sqrt{b^2-4ac}}{2a}, \quad \lambda^2 = \frac{b-\sqrt{b^2-4ac}}{2a}, \quad 0 < A_m, \quad 0 < \lambda^2 < \rho^2$
φ,k	$\varphi = \frac{1}{k_m} \ln\left(\rho\frac{\mathrm{sn}\left(\sqrt{a}\lambda\tau,k\right)}{\mathrm{cn}\left(\sqrt{a}\lambda\tau,k\right)}\right), \quad k = \sqrt{\frac{\rho^2-\lambda^2}{\rho}}$
A_m,A_a	$A_m(\varphi) = \frac{1}{A_a(\varphi)} = \rho\frac{\mathrm{sn}\left(\sqrt{a}\lambda\tau,k\right)}{\mathrm{cn}\left(\sqrt{a}\lambda\tau,k\right)}$

Brans-Dicke gravity solution : case 5 : $\varepsilon_1 = -1, \varepsilon_2 = -1, E_\varphi \in \mathbb{R}_-^*$	
a,b,c	$a \in \mathbb{R}_+^*, \quad b \in \mathbb{R}_-^*, \quad c \in \mathbb{R}_+^*, \quad b^2 > 4ac$
ρ,λ	$\rho^2 = \frac{-b-\sqrt{b^2-4ac}}{2a}, \quad \lambda^2 = \frac{-b+\sqrt{b^2-4ac}}{2a}, \quad 0 < \rho < \lambda < A_m$
φ,k	$\varphi = \frac{1}{k_m} \ln\left(\sqrt{\frac{\rho^2\mathrm{sn}^2(\sqrt{a}\lambda\tau,k)-\lambda^2}{\mathrm{sn}^2\left(\sqrt{a}\lambda\tau,k\right)-1}}\right), \quad k = \frac{\rho}{\lambda}$
A_m,A_a	$A_m(\varphi) = \frac{1}{A_a(\varphi)} = \sqrt{\frac{\rho^2\mathrm{sn}^2(\sqrt{\lvert a\rvert}\lambda\tau,k)-\lambda^2}{\mathrm{sn}^2\left(\sqrt{a}\lambda\tau,k\right)-1}}$

Figure 76 – Solutions analytiques du modèle de Brans-Dicke étudié. Cinq cas particuliers sont discernables à partir des valeurs de ε_1 et de ε_2, conduisant à des solutions analytiques exactes pour la forme de φ et de $A_m(\varphi)$. Ici, ρ et λ sont les racines du polynôme $ax^4 + bx^2 + c$ de l'équation A.10. La plupart des solutions s'expriment comme des combinaisons des fonctions elliptiques de Jacobi sn et cn (voir équation A.11).

plutôt, si, à condition d'y appliquer la bonne transformation à ce dernier. En effet, a_{JF} est lié à a_{EF} par :

$$\boxed{a_{JF}\left(t_{JF}\right) = A_m\left(\varphi\left(t_{EF}\left(t_{JF}\right)\right)\right) a_{EF}\left(t_{EF}\left(t_{JF}\right)\right)} \tag{A.12}$$

où la dépendance entre le temps t_{EF} dans la formulation d'Einstein et le temps t_{JF} dans la formulation de Jordan est donnée par :

$$t_{JF} - t_{JF_0} = \int_{t_{EF_0}}^{t_{EF}} A_m(\varphi\left(t\right)) \, \mathrm{d}t \tag{A.13}$$

Autrement dit, pour un temps $t = t_{EF}$ dans le référentiel d'Einstein donné, on peut calculer d'une part $a_{EF}\left(t\right)$ et d'autre part $\tau\left(t\right)$ donc $\varphi\left(\tau\right)$ donc $A_m\left(\varphi\right)$. On peut donc calculer d'une part le facteur d'échelle a_{JF} dans le référentiel de Jordan et d'autre part le temps t_{JF} correspondant, ce qui permet finalement de reconstruire la dynamique $a_{JF}\left(t_{JF}\right)$. Sauf que même en connaissant l'expression analytique de $A_m\left(\varphi\right)$ pour les différents cas présentés en figure 76, il n'existe pas d'expression simple pour l'intégration du terme droite de A.12. Le mieux que l'on puisse faire, c'est donner une expression approchée à l'origine dans le cas 3, c'est-à-dire lorsque $\varepsilon_1 = +1$ et $\varepsilon_2 = -1$ et pour des conditions très spécifiques[43] :

$$a_{JF}\left(t_{JF}\right) \simeq \sqrt[3]{3}\lambda \cosh\left(\frac{4t_{JF} + \lambda - \lambda \mathrm{W}\left(e^{\frac{\lambda + 4t_{JF}}{\lambda}}\right)}{2\lambda}\right) \exp\left(\frac{4t_{JF} + \lambda - \lambda \mathrm{W}\left(e^{\frac{\lambda + 4t_{JF}}{\lambda}}\right)}{6\lambda}\right) \tag{A.14}$$

avec :

$$\mathrm{W}\left(z\right) e^{\mathrm{W}(z)} = z, \qquad \lambda^2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \qquad \text{et} \qquad \begin{cases} a \ll 1, & b > 0, & c \ll 1 \\ E_\varphi > 0, & C_m \ll 1, & C_a \gg 1 \end{cases} \tag{A.15}$$

où W désigne la fonction W de Lambert[145].

A.5 Solutions numériques

Toutefois, le numérique va permettre d'aller plus loin et va permettre d'explorer les différentes solutions dans le référentiel de Jordan. Dans cette partie, on pose $a_0 = 1$ et $\varphi_0 = 0$. L'objectif est de résoudre numériquement l'équation A.12. Pour cela, un algorithme a été conçu et implémenté en C++11. Pour paver efficacement l'espace des paramètres k_m, C_m, C_a, E_φ, ε_1, ε_2 sur des machines multicœurs à mémoire partagée, une parallélisation triviale à base de threads de la librairie standard a été implémentée, chaque thread s'occupant alors d'un ensemble de paramètres donné. L'une des difficultés originellement rencontrée était le manque de stabilité de l'intégrateur : suivant les cas, de petites variations de t pouvaient entraîner de grandes variations de φ ou de A_m si bien que les solutions accumulaient rapidement des erreurs à proximité de certains points. Un contrôle du pas basé sur la conservation de E_φ ne convenait pas non plus car, sans vision globale de l'évolution de $V\left(\varphi\right)$, des cas pathologiques étaient rencontrés et n'offraient comme seuls choix que de diviser le pas de temps jusqu'à la précision machine ou de commencer à accumuler des erreurs non négligeables. En résumé, chercher à résoudre A.12 par intégration de t_{EF} « à l'aveugle », c'est-à-dire sans se servir de l'information sur le profil de $V\left(\varphi\right)$, était voué à l'échec.

Pour résoudre ce problème, la stratégie suivante a été adoptée : avant la phase d'intégration, le profil de $V\left(\varphi\right)$ est échantillonné pour les valeurs k_m, C_m et C_a du modèle en question. Cela commence par une recherche du plus petit φ_{min} et du plus grand φ_{max} tels que pour tout $\varphi \in [\varphi_{min}, \varphi_{max}]$ avec $\varphi_{min} \leq \varphi_0 \leq \varphi_{max}$, $\mathrm{isfinite}_{\mathrm{IEEE754}}\left(V\left(\varphi\right)\right)$ est vérifié. Autrement dit, cela donne l'intervalle autour de φ_0, pour lequel $V\left(\varphi\right)$ peut être calculé sans risque d'obtenir un résultat infini ou indéfini. Dans la pratique, cette recherche de limites de l'intervalle repose sur une simple méthode dichotomique sur

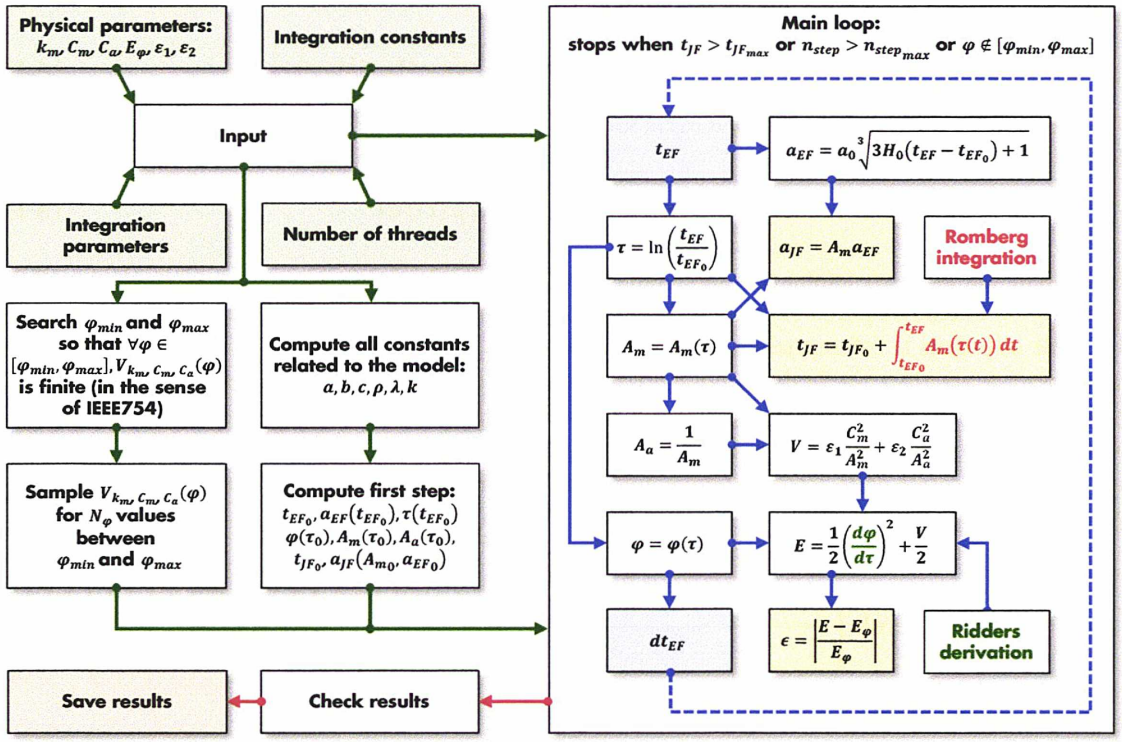


FIGURE 77 – Algorithme de résolution numérique de la dynamique dans le Jordan Frame. L'algorithme est composé de trois phases : une phase d'initialisation (flèches vertes), une phase de calcul (flèches bleues) et une phase de finalisation (flèches rouges). Une fois la consistance des données initiales contrôlée, l'algorithme commence par rechercher les bornes du domaine de validité de φ puis par échantillonner $V(\varphi)$ sur cet intervalle. Parallèlement à cela, les constantes et les données initiales du problème sont calculées. À l'issue de quoi la boucle principale est exécutée : chaque étape correspond à un temps t_{EF} donné à partir duquel sont calculées indépendamment le facteur d'échelle a_{JF} et le temps t_{JF} dans le Jordan Frame. A chaque étape, l'énergie est mesurée pour évaluer la précision et la stabilité du calcul. Le pas de temps $\mathrm{d}t_{EF}$ est adaptatif et est basé sur l'échantillonnage de $V(\varphi)$: l'écart entre deux pas de temps successif doit correspondre à un écart d'une subdivision dans l'échantillonnage. Une fois une condition d'arrêt atteinte, la boucle se termine, la consistance des résultats est contrôlée et les données sont enregistrées.

les valeurs flottantes de φ. Une fois les frontières du domaine trouvées, le profil de $V(\varphi)$ est échantillonné linéairement en fonction de φ : 100 000 ou 1 000 000 valeurs sont quasi-immédiates à obtenir et n'impactent pas significativement le temps global de calcul.

Une fois les limites du domaine de définition et la forme de $V(\varphi)$ connus avec un pas $\mathrm{d}\varphi$, l'évaluation numérique de la solution peut être effectuée sans risque de voir se propager les erreurs. Pour cela, une méthode adaptative basée sur l'échantillonnage préalable de $V(\varphi)$ est utilisée : à chaque étape du calcul $\mathrm{d}t_{EF}$ est augmenté ou diminué récursivement de sorte que la condition suivante soit vérifiée :

$$\mathrm{d}\varphi \leq |\varphi(t_{EF}) - \varphi(t_{EF} + \mathrm{d}t_{EF})| < 2\,\mathrm{d}\varphi \tag{A.16}$$

À partir de là, il n'y a plus de grande difficulté d'implémentation de la résolution de A.12 : à chaque pas de temps t_{EF}, les quantités a_{EF}, τ, φ, A_m, A_a, V sont calculées par les expressions analytiques présentées dans les paragraphes précédents. $a_{JF} = A_m a_{EF}$ est obtenu immédiatement et t_{JF} est calculé suivant l'expression A.13 par une intégration de Romberg[146]. À chaque pas de temps, l'énergie A.6 est également calculée pour évaluer la précision et la stabilité du calcul, la dérivée de φ par rapport à τ étant déterminée via une dérivation de Ridders[147]. Les fonctions elliptiques de Jacobi, elles, sont calculées au moyen d'une version optimisée de l'algorithme fourni dans Press *et al.* 2007[143] pour

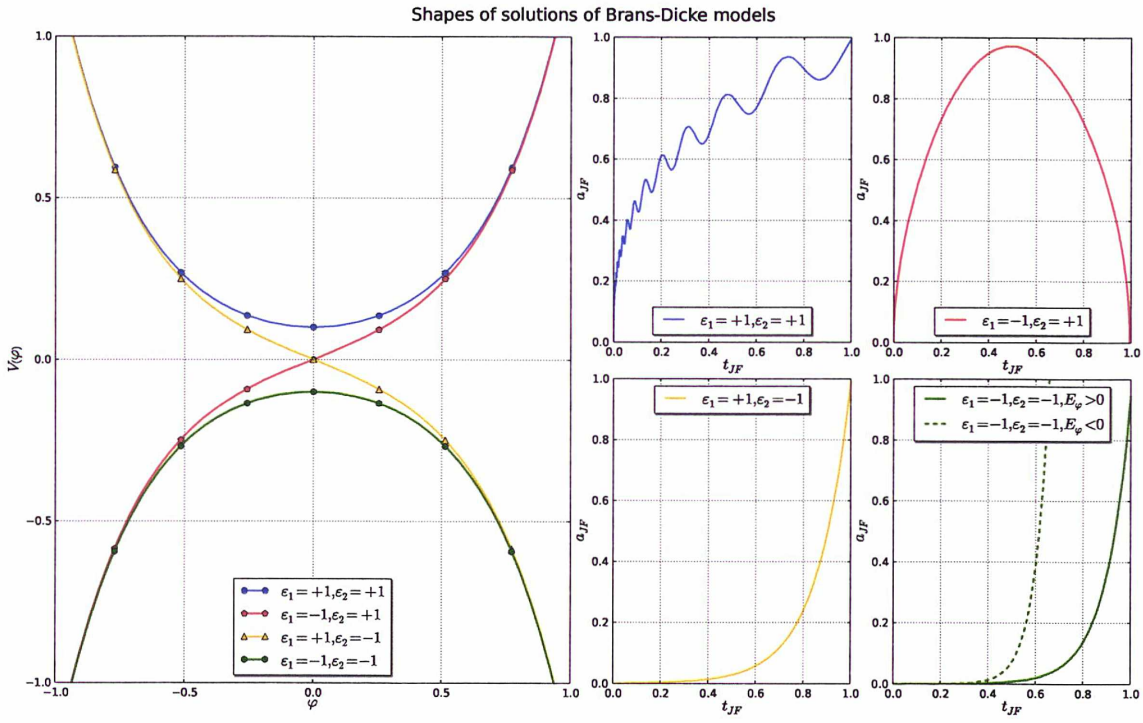


FIGURE 78 – Profil des solutions numériques du modèle de Brans-Dicke considéré. Les axes ont été renormalisés afin d'améliorer la lisibilité de la figure : il s'agit juste ici de mettre en évidence les comportements des différentes solutions exposées en figure 76. À gauche, les symétries de $V(\varphi)$ sont déterminées par les valeurs de ε_1 et ε_2 tandis que son profil est piloté par les valeurs de C_m, C_a, A_m et A_a. Ce profil est exploré avant la phase de calcul proprement dite par l'algorithme en vue de déterminer le domaine de validité des solutions et de minimiser la propagation d'erreurs. Il est alors possible d'explorer, en parallèle, les comportements généraux de l'évolution du facteur d'échelle dans le référentiel de Jordan pour les cinq types de solutions comme illustré dans les figures de droite. Le cas des ε positifs, correspondant à des termes cinétiques ordinaires pour la matière et le secteur AWE, permet d'obtenir des solutions d'expansion oscillante. Lorsque la matière est associée à un champ scalaire « phantom », l'Univers se recontracte rapidement après une phase initiale d'expansion. Les trois autres cas, c'est-à-dire lorsqu'un champ scalaire « phantom » est associé au secteur AWE, conduisent à l'obtention de profils très accélérées, potentiellement intéressants dans un contexte inflationnaire.

tirer parti du fait que le module elliptique k est constant pour un modèle donné : à chaque appel de l'algorithme, certaines variables locales sont conservées en mémoire par chaque thread actif pour éviter des calculs inutiles et possiblement coûteux à l'appel suivant. L'algorithme global est résumé en figure 77. Il permet au final, pour chaque ensemble de paramètres du modèle, d'obtenir les valeurs de a_{JF} et t_{JF} et donc l'évolution du facteur d'échelle $a_{JF}(t_{JF})$.

A.6 Résultats

En utilisant l'algorithme décrit dans les paragraphes précédents aux solutions analytiques exposées en figure 76, il est possible d'explorer numériquement l'espace des paramètres, là où la dérivation d'expressions analytiques était impossible. Cela permet d'obtenir des profils de solutions très divers comme présenté en figure 78.

Ces différents comportements sont pilotés par les signes des termes cinétiques du champ scalaire associé à la matière et du champ scalaire associé à secteur AWE. Certaines solutions sont ainsi oscillantes, d'autres se terminent rapidement après une phase de recontraction et d'autres enfin peuvent exhiber des accélérations de l'évolution du facteur d'échelle dans le référentiel de Jordan très importantes. Pour ces derniers types de solutions, l'évaluation préalable des bornes du domaine de validité

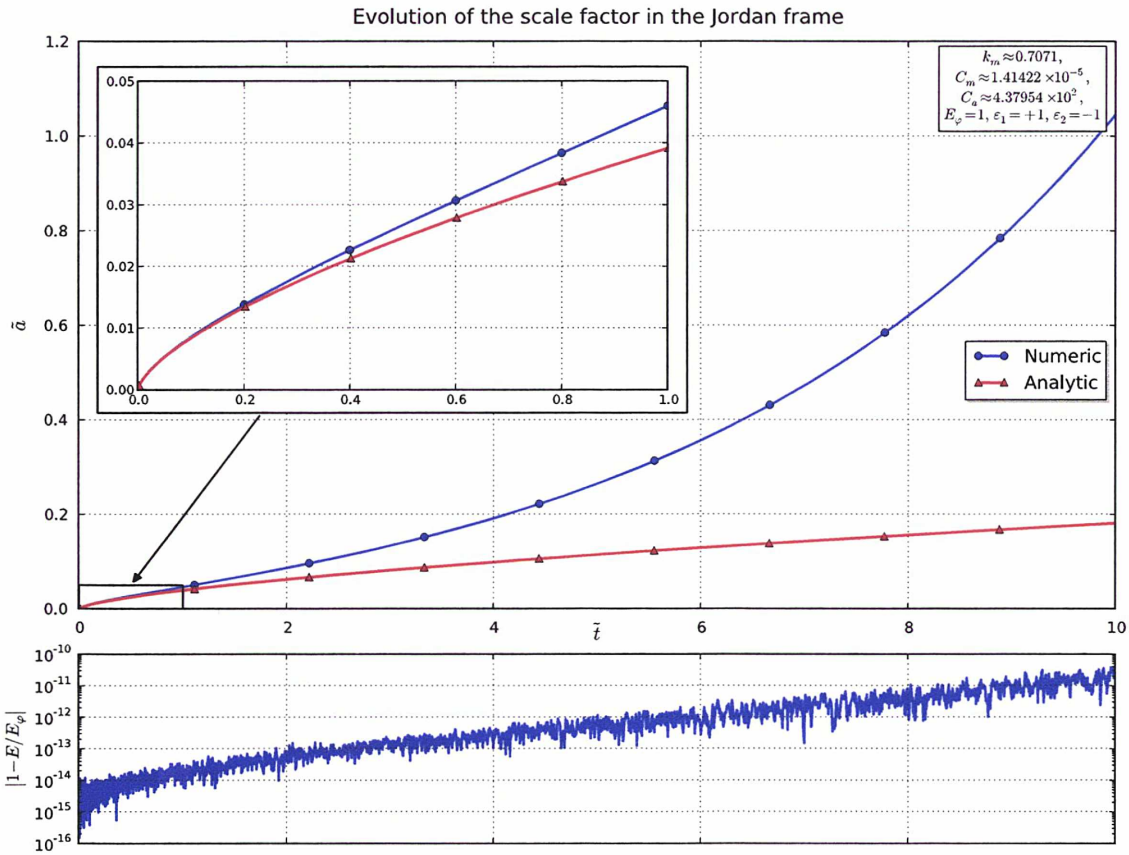


Figure 79 – Comparaison de la solution analytique et de la solution numérique pour le troisième type de solution dans le Jordan Frame. Dans la partie supérieure de la figure, les différences entre la solution analytique A.14 et la solution numérique pour le même ensemble de paramètres sont mises en évidence. Ces paramètres ont été choisis en vue de minimiser l'erreur commise par la solution analytique en fonctions de Lambert. Toutefois, comme on peut le voir, très rapidement la solution numérique s'éloigne de l'approximation analytique : et c'est normal car la première n'est valide qu'à l'origine. Sur de petites échelles de temps, comme illustré dans l'encadré, les deux se comportent de la même façon. Cela permet entre autres de vérifier qualitativement, pour ce cas au moins, la validité de l'algorithme. Sur des durées plus importantes, la forme analytique n'est pas à même de reproduire un comportement accéléré. Dans ce cas, seul le numérique permet d'explorer la dynamique de ce type de modèles. La qualité de la solution numérique peut également être vérifiée en contrôlant l'erreur relative sur la conservation de E_φ. Comme cela est illustré dans la partie inférieure de la figure, celle-ci reste ici inférieure à 10^{-10}, ce qui permet de garantir la validité des résultats.

φ_{min} et φ_{max} sont essentielles de même que l'échantillonnage de $V(\varphi)$ pour contrôler le pas de temps adaptatif. La comparaison du résultat numérique au résultat analytique en fonctions de Lambert présenté en figure 79 pour un ensemble de paramètres compatibles avec l'approximation de l'équation A.14 permet de s'assurer de la validité de l'algorithme. De plus, le contrôle de l'erreur sur E_φ tout au long du calcul permet de vérifier la qualité de la solution. Pour de très fortes accélérations cette erreur augmente progressivement, mais dans ce cas, φ sort de l'intervalle $[\varphi_{min}, \varphi_{max}]$ et l'algorithme s'arrête avant que l'erreur n'impacte significativement les résultats. Autrement dit a_{JF} atteint l'infini au sens de la représentation des nombres flottants avant que l'erreur sur a_{JF} n'ait le temps de se propager.

Au-delà de l'analyse qualitative des solutions du modèle, il est possible de tester la dépendance d'un type de solution particulier aux paramètres k_m, C_m, C_m, et E_φ. C'est ce qui est fait en figures 80 et 81 dans le cas où $\varepsilon_1 = +1$ et où $\varepsilon_2 = -1$. Comme attendu, une augmentation de la constante de couplage k_m produit des accélérations significativement plus importantes. Aussi, des accélérations de type inflationnaires pourraient être aisément produites sur des temps très courts dans le cadre de solutions où le secteur AWE est de type « phantom ». À l'opposé, des accélérations légères beaucoup

plus tardives pourraient être produites dans le Jordan Frame et seraient donc à même de reproduire des comportements caractéristiques de l'énergie noire.

A.7 Conclusion

Il existe de multiples manières de modéliser une accélération de l'expansion au-delà de la constante cosmologique sur diverses bases théoriques. La quintessence et la théorie de Brans-Dicke sont deux de ces approches. Cette dernière, dans le cadre de l'hypothèse AWE, permet un traitement unifié du secteur sombre. Une grande variété de profils d'expansion peut être obtenue en mêlant les approches analytiques et numériques comme illustré dans cette dernière section, l'algorithme développé à cette occasion permettant de paver finement l'espace des paramètres.

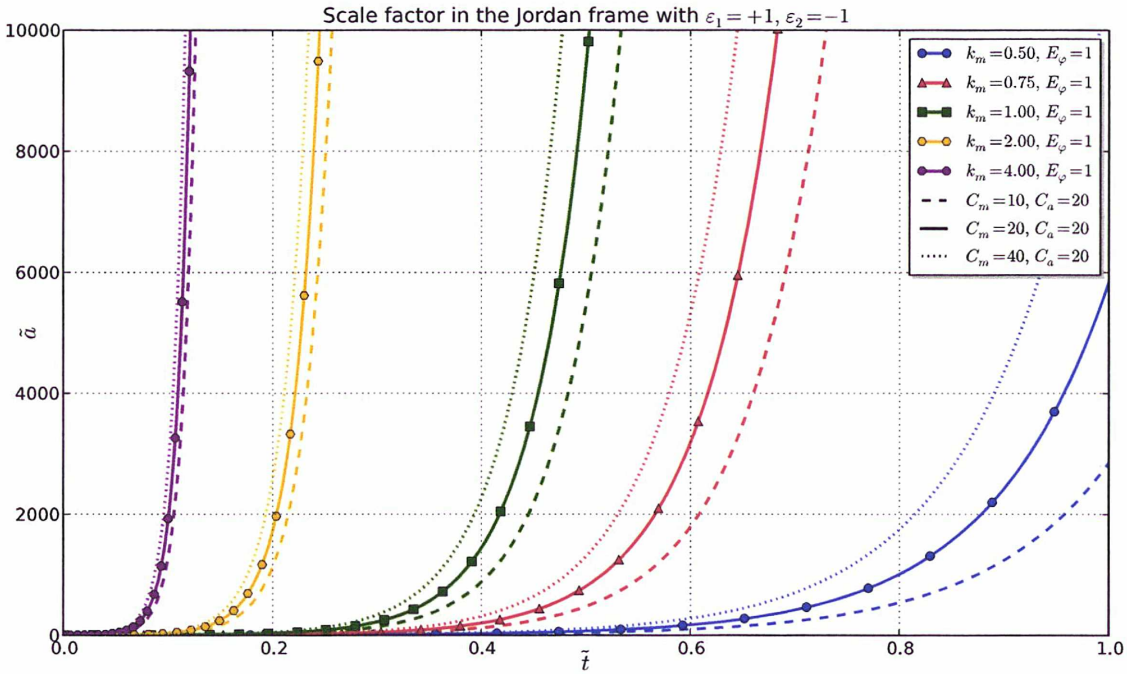


FIGURE 80 – Evolution du facteur d'échelle dans le Jordan Frame pour le troisième type de solution. Dans le cas où la matière est associée à un champ scalaire ordinaire, et où le secteur invisible est associé à un champ scalaire de type « phantom », des profils très accélérés peuvent être mis en évidence. Ici, quinze jeux de paramètres sont illustrés afin de montrer la dépendance de la dynamique du facteur d'échelle dans le Jordan Frame vis-à-vis de k_m, C_m, C_m, et E_φ. Pour le jeu de paramètres choisi, une augmentation du paramètre de couplage k_m, ou de C_m, se traduit par une accélération de l'expansion plus importante.

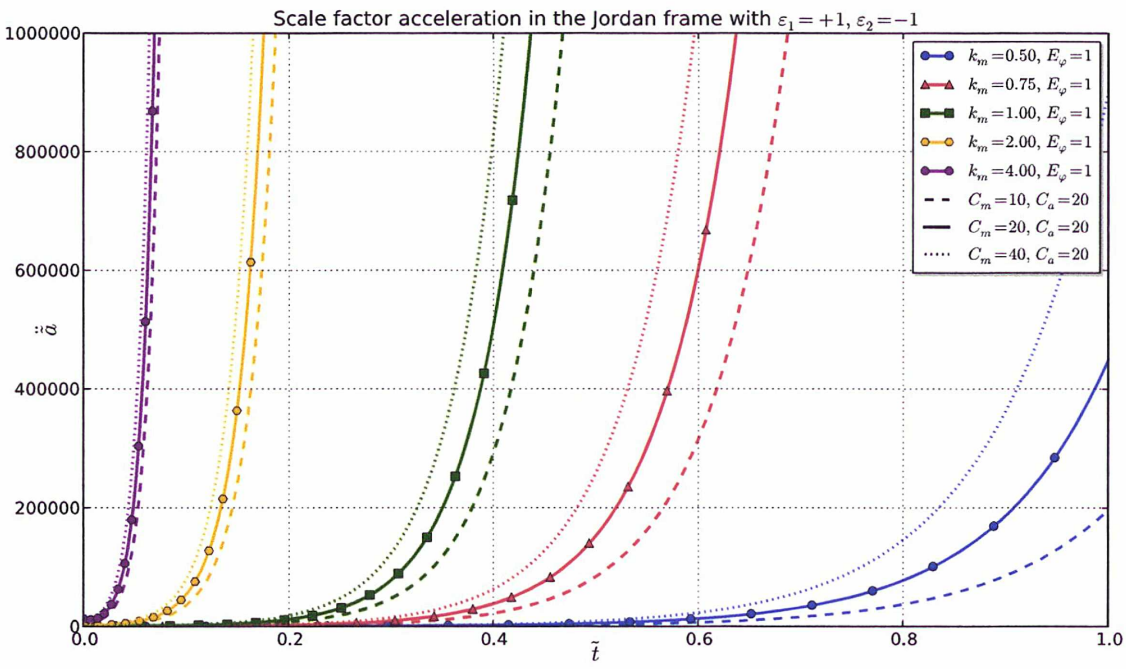


FIGURE 81 – Accélération du facteur d'échelle dans le Jordan Frame pour le troisième type de solution. Les valeurs des accélérations du facteur d'échelle sont représentées pour les mêmes jeux de paramètres que ceux de la figure 80. En augmentant encore k_m ou C_m, de très fortes accélérations pourraient être obtenues sur des temps très courts pour obtenir des comportements inflationnaires. Pour ce genre de profils, la phase initiale de détermination du profil de $V(\varphi)$ est primordiale pour minimiser les erreurs et contrôler le pas adaptatif de l'intégration.

Annexe B

Géodésiques nulles en métrique de Schwarzschild

***

B.1 Expressions en termes d'intégrales 189
B.2 Solutions en fonctions de Weierstrass 193

B.1 Expressions en termes d'intégrales

Soit une masse sphérique M statique de rayon de Schwarzschild r_s au sein d'un Univers vide, la métrique de Schwarzschild en coordonnées sphériques (t, r, θ, ϕ) s'écrit :

$$\boxed{\mathrm{d}s^2 = \left(1 - \frac{r_s}{r}\right) c^2\,\mathrm{d}t^2 - \frac{\mathrm{d}r^2}{\left(1 - \frac{r_s}{r}\right)} - r^2\left(\mathrm{d}\theta^2 + \sin^2(\theta)\,\mathrm{d}\phi^2\right)} \tag{B.1}$$

avec :

$$\boxed{r_s = \frac{2GM}{c^2}} \tag{B.2}$$

Dans toute la suite, on ne s'intéresse qu'aux géodésiques nulles, c'est-à-dire avec $\mathrm{d}s^2 = 0$ et l'on se place, de plus, dans le plan de propagation $\theta = \frac{\pi}{2}$. Dans ce cas l'équation B.1 peut se réécrire, en posant $A(r) = \left(1 - \frac{r_s}{r}\right)$:

$$A(r)\,c^2\,\mathrm{d}t^2 - \frac{\mathrm{d}r^2}{A(r)} - r^2\,\mathrm{d}\phi^2 = 0 \quad \Rightarrow \quad \frac{1}{r^4}\left(\frac{\mathrm{d}r}{\mathrm{d}\phi}\right)^2 + \frac{A(r)}{r^2} = \frac{A(r)^2\,c^2}{r^4}\left(\frac{\mathrm{d}t}{\mathrm{d}\phi}\right)^2 \tag{B.3}$$

L'équation B.1 dans le plan $\theta = \frac{\pi}{2}$ peut également se réécrire sous la forme :

$$\left(\frac{\mathrm{d}s}{\mathrm{d}\lambda}\right)^2 = g_{\mu\nu}\frac{\mathrm{d}x^\mu}{\mathrm{d}\lambda}\frac{\mathrm{d}x^\nu}{\mathrm{d}\lambda} = A(r)\,c^2\left(\frac{\mathrm{d}t}{\mathrm{d}\lambda}\right)^2 - \frac{1}{A(r)}\left(\frac{\mathrm{d}r}{\mathrm{d}\lambda}\right)^2 - r^2\left(\frac{\mathrm{d}\phi}{\mathrm{d}\lambda}\right)^2 \tag{B.4}$$

que l'on peut directement injecter dans l'équation d'Euler-Lagrange, où $\dot{x^\sigma} = \frac{\mathrm{d}x^\sigma}{\mathrm{d}\lambda}$:

$$\frac{\mathrm{d}}{\mathrm{d}\lambda}\frac{\partial}{\partial \dot{x}^\sigma}\sqrt{|g_{\mu\nu}\dot{x}^\mu\dot{x}^\nu|} = \frac{\partial}{\partial x^\sigma}\sqrt{|g_{\mu\nu}\dot{x}^\mu\dot{x}^\nu|} \tag{B.5}$$

ce qui donne, pour $\sigma = t$ et $\sigma = \phi$:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\lambda}\left[\frac{A(r)\,c^2\frac{\mathrm{d}t}{\mathrm{d}\lambda}}{\sqrt{|g_{\mu\nu}\dot{x}^\mu\dot{x}^\nu|}}\right] = 0 \qquad &\Rightarrow \qquad A(r)\frac{\mathrm{d}t}{\mathrm{d}\lambda} = \frac{a}{b} \\ \frac{\mathrm{d}}{\mathrm{d}\lambda}\left[\frac{-r^2\frac{\mathrm{d}\phi}{\mathrm{d}\lambda}}{\sqrt{|g_{\mu\nu}\dot{x}^\mu\dot{x}^\nu|}}\right] = 0 \qquad &\Rightarrow \qquad r^2\frac{\mathrm{d}\phi}{\mathrm{d}\lambda} = ac \end{aligned} \tag{B.6}$$

où a et b sont les deux constantes du mouvement souvent utilisées en métrique de Schwarzschild. De ces expressions, on déduit :

$$\frac{A(r)^2\,c^2}{r^4}\left(\frac{\mathrm{d}t}{\mathrm{d}\phi}\right)^2 = \left(\frac{A(r)\frac{\mathrm{d}t}{\mathrm{d}\lambda}}{\frac{r^2}{c}\frac{\mathrm{d}\phi}{\mathrm{d}\lambda}}\right)^2 = \frac{1}{b^2} \tag{B.7}$$

ce qui permet de réécrire l'équation B.3 sous la forme :

$$\frac{1}{r^4}\left(\frac{\mathrm{d}r}{\mathrm{d}\phi}\right)^2 + \frac{A(r)}{r^2} = \frac{1}{b^2} \qquad \Rightarrow \qquad \left(\frac{\mathrm{d}r}{\mathrm{d}\phi}\right)^2 = r^4\left(\frac{1}{b^2} - \frac{1}{r^2}\left(1 - \frac{r_s}{r}\right)\right) \tag{B.8}$$

Cette quantité est constante pour tout r lorsque sa dérivée par rapport à r est nulle, c'est-à-dire lorsque :

$$\frac{\mathrm{d}}{\mathrm{d}r}\left[\left(\frac{\mathrm{d}r}{\mathrm{d}\phi}\right)^2\right] = \frac{\mathrm{d}}{\mathrm{d}r}\left[r^4\left(\frac{1}{b^2} - \frac{1}{r^2}\left(1 - \frac{r_s}{r}\right)\right)\right] = \frac{4r^3}{b^2} - 2r + r_s \tag{B.9}$$

s'annule. Comme de plus, la dérivée de r par rapport à ϕ est égale à 0 lorsque :

$$\frac{1}{b^2} = \frac{1}{r^2}\left(1 - \frac{r_s}{r}\right) = \frac{r - r_s}{r^3} \tag{B.10}$$

le rayon critique r_c correspondant à une orbite circulaire vaut :

$$\frac{4r_c^3}{b^2} - 2r_c + r_s = \frac{4r_c^3}{r_c^2}\left(1 - \frac{r_s}{r_c}\right) - 2r_c + r_s = 0 \qquad \Rightarrow \qquad \boxed{r_c = \frac{3}{2}r_s} \tag{B.11}$$

Dans toute la suite, on s'intéresse aux orbites non-liées, c'est-à-dire pour lesquelles $r > \frac{3}{2}r_s$ ou encore $b > b_c$ avec :

$$b_c = \sqrt{\frac{r_c^3}{r_c - r_s}} = \frac{3\sqrt{3}}{2}r_s \tag{B.12}$$

Pour ces orbites, la dérivée de r par rapport à ϕ s'annule au point d'approche maximale r_0 avec $b > r_0 > r_s$, qui est, par conséquent, solution de :

$$\frac{1}{b^2} - \frac{1}{r^2}\left(1 - \frac{r_s}{r}\right) = 0 \qquad \Rightarrow \qquad r^3 - b^2 r + b^2 r_s = 0 \tag{B.13}$$

Or d'après les formules de Viète[148], en supposant $p \in \mathbb{R}^*$, $q \in \mathbb{R}$ et $z \in \mathbb{C}$, en notant (z_i, z_j, z_k) les trois racines du polynôme cubique et en posant $\Delta = -4p^3 - 27q^2$, on a :

$$z^3 + pz + q = 0 \quad \begin{cases} \Delta > 0 \;\Rightarrow\; (z_i, z_j, z_k) \in \mathbb{R}^3 & \text{et} \quad z_i \neq z_j, \quad z_j \neq z_k \\ \Delta = 0 \;\Rightarrow\; (z_i, z_j, z_k) \in \mathbb{R}^3 & \text{et} \quad z_i \neq z_j, \quad z_j = z_k \\ \Delta < 0 \;\Rightarrow\; (z_i, z_j, z_k) \in \mathbb{R} \times \mathbb{C}^2 & \text{et} \quad z_i \neq z_j, \quad z_j = \bar{z}_k \end{cases} \tag{B.14}$$

avec :

$$z_n = 2\sqrt{-\frac{p}{3}}\cos\left(\frac{1}{3}\cos^{-1}\left(\frac{3q}{2p}\sqrt{-\frac{3}{p}}\right) - \frac{2n\pi}{3}\right) \qquad \text{pour} \qquad n = 0, 1, 2 \tag{B.15}$$

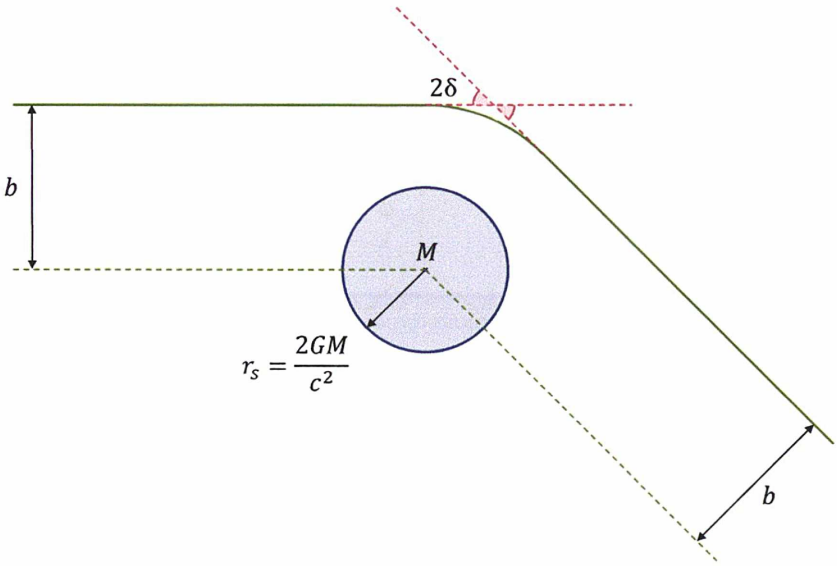


Figure 82 – Géodésique nulle en métrique de Schwarzschild. On considère un photon provenant de l'infini avec un paramètre d'impact b en direction d'une masse M et l'on s'intéresse à l'angle total de déflexion 2δ.

Pour $p = -b^2$, $q = b^2 r_s$ et $b > b_c$, Δ est strictement positif et l'on obtient finalement la relation suivante entre b et r_0, la plus grande des racines :

$$\boxed{b = \sqrt{\frac{r_0^3}{r_0 - r_s}} \quad \Leftrightarrow \quad r_0 = \frac{2b}{\sqrt{3}} \cos\left(\frac{1}{3}\cos^{-1}\left(-\frac{3\sqrt{3}r_s}{2b}\right)\right)} \tag{B.16}$$

où b représente le paramètre d'impact tel qu'illustré en figure 82. A partir de ces paramètres, le problème peut être reformulé en termes légèrement différents, bien que strictement équivalents à ce qui a été présenté jusqu'à maintenant. Au lieu de considérer un photon provenant de l'infini avec un paramètre d'impact b, on peut considérer un photon « lancé » d'une distance $r_0 > r_c$ d'une masse sphérique statique et s'éloignant vers l'infini avec le paramètre d'impact b, relié à r_0 par l'équation B.16. Le problème global est alors rendu symétrique par rapport à r_0, comme illustré par le schéma présenté en figure 83.

A partir de l'équation B.8, il est possible de réécrire l'équation de la trajectoire sous la forme couramment utilisée :

$$\boxed{\phi(r) = \int_{r_0}^{r} \frac{\mathrm{d}\rho}{\rho^2 \sqrt{\frac{1}{b^2} - \frac{1}{\rho^2}\left(1 - \frac{r_s}{\rho}\right)}}} \tag{B.17}$$

L'angle total de déflexion à l'infini vaut alors :

$$2\delta_\infty = 2 \lim_{r \to +\infty} \phi(r) = 2\int_{r_0}^{+\infty} \frac{\mathrm{d}\rho}{\rho^2 \sqrt{\frac{1}{b^2} - \frac{1}{\rho^2}\left(1 - \frac{r_s}{\rho}\right)}} = 2\int_0^{\frac{1}{r_0}} \frac{\mathrm{d}u}{\sqrt{\frac{1}{b^2} - u^2(1 - u r_s)}} \tag{B.18}$$

Cette dernière expression peut se réécrire de plusieurs façons sous la forme d'intégrales elliptiques et

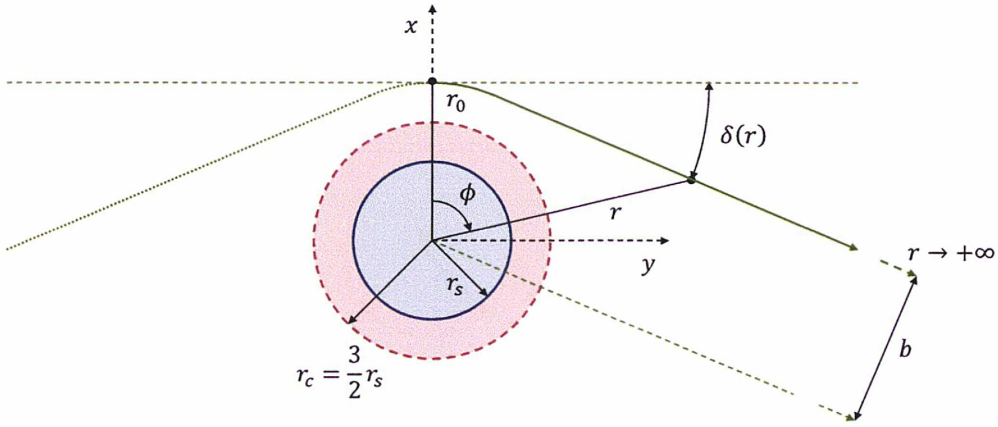


FIGURE 83 – Illustration du problème symétrisé par rapport à la distance minimale r_0. On peut considérer de façon équivalente une particule sans masse « tirée » du point $(r_0, 0)$ et s'éloignant asymptotiquement vers le paramètre d'impact b. Pour tout point de la trajectoire $\phi(r)$, il est possible de définir un demi-angle de déflexion $\delta(r)$.

notamment[149] :

$$\boxed{2\delta_\infty = 4\sqrt{\frac{r_0}{q}}\left(F\left(\frac{\pi}{2}, k\right) - F\left(\sigma_0, k\right)\right) - \pi} \quad \text{avec} \quad \begin{cases} F(\sigma, k) & = \int_0^\sigma \frac{\mathrm{d}\theta}{\sqrt{1-k^2\sin^2(\theta)}} \\ q & = \sqrt{r_0^2 + 2r_0 r_s - 3r_s^2} \\ k & = \sqrt{\frac{q - r_0 + 3r_s}{2q}} \\ \sigma_0 & = \sin^{-1}\left(\sqrt{\frac{q - r_0 + r_s}{q - r_0 + 3r_s}}\right) \end{cases} \tag{B.19}$$

où F correspond à une intégrale elliptique incomplète du premier type. Dans l'approximation de champ faible, on considère que b est très supérieur au rayon de Schwarzschild, c'est-à-dire que $\varepsilon = \frac{r_s}{b} \ll 1$. Dans ce cas :

$$\begin{cases} \frac{r_s}{r_0} = \frac{\sqrt{3}\varepsilon}{2} \frac{1}{\cos\left(\frac{1}{3}\cos^{-1}\left(-\frac{3\sqrt{3}\varepsilon}{2}\right)\right)} \simeq \varepsilon + \mathcal{O}\left(\varepsilon^2\right) \\ 4\sqrt{\frac{r_0}{q}} \simeq 4 - 2\varepsilon + \mathcal{O}\left(\varepsilon^2\right) \\ F\left(\frac{\pi}{2}, k\right) \simeq \frac{\pi}{2}\left(1 + \frac{\varepsilon}{2}\right) + \mathcal{O}\left(\epsilon^2\right) \\ F(\sigma, k) \simeq \frac{\pi}{4}\left(1 + \frac{\varepsilon}{2}\right) - \frac{\varepsilon}{2} + \mathcal{O}\left(\varepsilon^2\right) \end{cases} \tag{B.20}$$

si bien qu'au premier ordre :

$$\boxed{2\delta_\infty \simeq 2\varepsilon + \mathcal{O}\left(\varepsilon^2\right) = \frac{4GM}{c^2 b} + \mathcal{O}\left(\varepsilon^2\right)} \tag{B.21}$$

ce qui correspond à la formule des lentilles couramment utilisée dans nombre d'applications lié au weak-lensing. Dans l'exemple classique du Soleil, c'est-à-dire pour rayon lancé à $b = r_\odot$ d'une masse $M = M_\odot$, la précision relative de la formule des lentilles est de l'ordre de 6.8×10^{-6}. Pour un halo homogène de masse $M = 10^{14} M_\odot$ avec $b = r_{200} = \sqrt[3]{\frac{M}{\frac{4}{3}\pi \times 200 \rho_c}}$, la précision relative tombe à environ 1.5×10^{-5}, ce qui rend cette formule inutilisable en pratique pour évaluer des erreurs numériques telles que celles de l'algorithme de raytracing décrit au dernier chapitre.

L'équation B.19, bien qu'exacte, est inutilisable en pratique pour évaluer des erreurs numériques car elle ne donne que l'angle de déviation à l'infini. L'objectif est ici de dériver l'expression analytique de toute la trajectoire, ou plutôt celle de l'angle de déviation $\delta(r)$ tout le long de cette dernière comme illustré en figure 83. L'expression B.17 de $\phi(r)$ en termes d'intégrale pourrait convenir si toutefois elle ne divergeait pas numériquement en $r = r_0$, introduisant une potentielle source d'erreurs numériques.

Erreurss minime sans doute, mais erreurs quand même. Pour contourner ce problème, on se propose dans la section suivante de déterminer une expression exacte de $\delta(r)$ en termes de fonctions spéciales.

B.2 Solutions en fonctions de Weierstrass

Dans toute cette partie, on pose :

$$\alpha = \frac{r_s}{r_0} = \beta + \frac{1}{3} \qquad \text{et} \qquad \mu = \frac{r_s}{r} = \nu + \frac{1}{3} \tag{B.22}$$

Par changement de variable, l'intégrale B.17 peut se réécrire :

$$\phi(r) = \int_{\frac{r_0}{r}}^{\alpha} \frac{\mathrm{d}\mu}{\sqrt{\alpha^2(1-\alpha) - \mu^2(1-\mu)}} = \int_{\frac{r_0}{r}-\frac{1}{3}}^{\beta} \frac{\mathrm{d}\nu}{\sqrt{\left(\nu^3 - \frac{\nu}{3}\right) - \left(\beta^3 - \frac{\beta}{3}\right)}} \tag{B.23}$$

A ce stade, on introduit la fonction elliptique de Weierstrass $\wp(z) = \wp(z, g_2, g_3)$ telle que :

$$\left(\frac{\mathrm{d}\wp(z)}{\mathrm{d}z}\right)^2 = 4\wp(z)^3 - g_2\wp(z) - g_3 \qquad \text{d'invariants} \qquad \begin{cases} g_2 = \frac{4}{3} \\ g_3 = 4\left(\beta^3 - \frac{\beta}{3}\right) \end{cases} \tag{B.24}$$

si bien que :

$$\begin{aligned} \phi(r) = \; & 2\int_{\frac{r_0}{r}-\frac{1}{3}}^{\beta} \frac{\mathrm{d}\wp(z)}{\sqrt{4\wp(z)^3 - g_2\wp(z) - g_3}} \\ = \; & 2\left(\int_{\frac{r_0}{r}-\frac{1}{3}}^{+\infty} \frac{\mathrm{d}\wp(z)}{\sqrt{4\wp(z)^3 - g_2\wp(z) - g_3}} - \int_{\beta}^{+\infty} \frac{\mathrm{d}\wp(z)}{\sqrt{4\wp(z)^3 - g_2\wp(z) - g_3}}\right) \end{aligned} \tag{B.25}$$

Or comme :

$$\Sigma = \int_{\sigma}^{+\infty} \frac{\mathrm{d}z}{\sqrt{4z^3 - g_2 z - g_3}} \qquad \Rightarrow \qquad \sigma = \wp(\Sigma) = \wp(-\Sigma) \tag{B.26}$$

on obtient :

$$\phi(r) = 2\left[\wp^{-1}(\beta) - \wp^{-1}\left(\frac{r_0}{r} - \frac{1}{3}\right)\right] \tag{B.27}$$

La simplification de cette équation passe par la détermination des racines (z_1, z_2, z_3) de $f(z) = 4z^3 - g_2 z - g_3$. Dans le cas le plus général, elles sont égales à :

$$\begin{cases} z_1 &= \frac{\zeta}{6} + \frac{g_2}{2\zeta} \\ z_2 &= -\frac{\zeta}{12} - \frac{g_2}{4\zeta} + i\frac{\sqrt{3}}{2}\left(\frac{\zeta}{6} - \frac{g_2}{2\zeta}\right) \\ z_3 &= -\frac{\zeta}{12} - \frac{g_2}{4\zeta} - i\frac{\sqrt{3}}{2}\left(\frac{\zeta}{6} - \frac{g_2}{2\zeta}\right) \end{cases} \qquad \text{avec} \qquad \zeta = \sqrt[3]{27g_3 + 3\sqrt{81g_3^2 - 3g_2^3}} \tag{B.28}$$

Sous la forme $g(z) = z^3 + pz + q = z^3 - \frac{g_2}{4}z - \frac{g_3}{4}$, ces mêmes racines sont données par la formule de Viète B.15 :

$$z_n = \sqrt{\frac{g_2}{3}}\cos\left(\frac{1}{3}\cos^{-1}\left(\frac{3\sqrt{3}g_3}{g_2\sqrt{g_2}}\right) - \frac{2n\pi}{3}\right) = \frac{2}{3}\cos\left(\frac{1}{3}\cos^{-1}\left(\frac{27}{2}\left(\beta^3 - \frac{\beta}{3}\right)\right) - \frac{2n\pi}{3}\right) \tag{B.29}$$

Comme ici $r_0 > \frac{3}{2}r_s$, on a $0 < \frac{r_s}{r_0} < \frac{2}{3}$ et $-\frac{1}{3} < \beta < \frac{1}{3}$ et les racines de g sont toutes réelles comme illustré en figure 84. Dans toute la suite, on pose :

$$e_1 > e_2 > e_3 \qquad \begin{cases} e_1 = z_0 = \wp(\omega_1) = +\frac{2}{3}\cos\left(\frac{1}{3}\cos^{-1}\left(\frac{27}{2}\left(\beta^3 - \frac{\beta}{3}\right)\right)\right) \\ e_2 = z_1 = \wp(\omega_2) = \beta \\ e_3 = z_2 = \wp(\omega_3) = -\frac{2}{3}\cos\left(\frac{1}{3}\cos^{-1}\left(\frac{27}{2}\left(\frac{\beta}{3} - \beta^3\right)\right)\right) \end{cases} \tag{B.30}$$

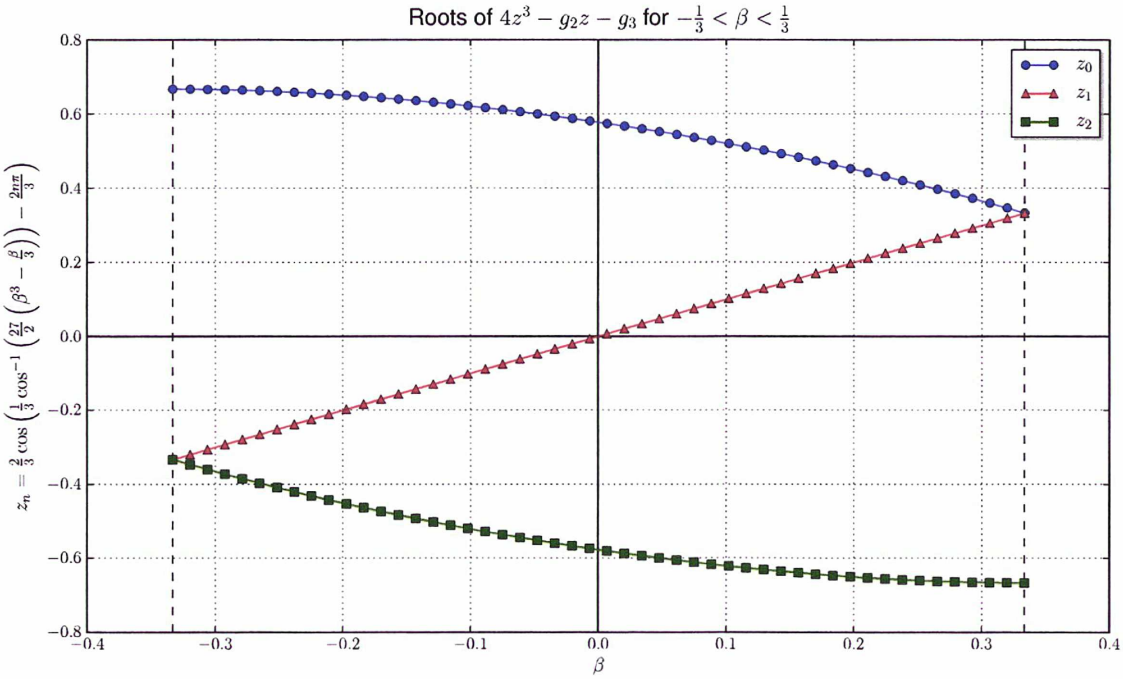


FIGURE 84 – Valeurs des racines du polynôme cubique impliqué dans la réécriture de $\phi(r)$ en fonctions de Weierstrass. Pour $-\frac{1}{3} < \beta < \frac{1}{3}$, les trois racines sont réelles et $z_0(\beta) > z_1(\beta) > z_2(\beta)$. Elles vérifient de plus $z_1(\beta) = \beta$ et $z_2(\beta) = -z_0(-\beta)$.

Par les formules d'addition et de parité des fonctions elliptiques de Weierstrass[150] :

$$\wp(x+y) = \frac{1}{4}\left(\frac{\wp'(x) - \wp'(y)}{\wp(x) - \wp(y)}\right)^2 - (\wp(x) + \wp(y)) \qquad \text{et} \qquad \begin{cases} \wp(-z) = \wp(z) \\ \wp'(-z) = -\wp'(z) \end{cases} \tag{B.31}$$

on obtient :

$$\begin{aligned} \wp(\omega_2 - z) &= \frac{1}{4}\left(\frac{\wp'(\omega_2) - \wp'(-z)}{\wp(\omega_2) - \wp(-z)}\right)^2 - (\wp(\omega_2) + \wp(-z)) \\ &= \frac{\wp(z)^3 - \frac{g_2}{4}\wp(z) - \frac{g_3}{4}}{(\beta - \wp(z))^2} - (\beta + \wp(z)) \\ &= \beta + \frac{9\beta^2 - 1}{3(\wp(z) - \beta)} \end{aligned} \tag{B.32}$$

ce qui, injecté dans B.27, donne finalement :

$$\boxed{\phi(r) = 2\wp^{-1}\left(\beta + \frac{9\beta^2 - 1}{3(\nu - \beta)}\right) = 2\wp^{-1}\left(\alpha - \frac{1}{3} + \frac{3\alpha(\alpha - 2)}{\frac{r_s}{r} - \alpha}\right) \qquad \text{avec} \qquad \alpha = \frac{r_s}{r_0}} \tag{B.33}$$

Annexe C

Articles et publications

***

C.1 Solution elliptiques en gravité généralisée de Brans-Dicke avec un couplage non-universel 195

C.2 Oscillations acoustiques des baryons limitées par la variance cosmique au sein de la simulation DEUS-FUR ΛCDM 196

C.3 Matrice de covariance du spectre de puissance de la matière dans les simulations DEUS-PUR ΛCDM : résolution en masse et erreurs non-gaussiennes 196

C.4 La première simulation à l'échelle de l'Univers Observable 197

C.5 DEUS Full Observable Universe Simulations : défi numérique et perspectives 197

Cette annexe rassemble les articles publiés au cours de cette thèse et apportant des détails et des éclairages sur certaines parties de ce manuscrit. Pour chaque article, le résumé est présenté, ainsi que la première page et un lien vers la source.

C.1 Solution elliptiques en gravité généralisée de Brans-Dicke avec un couplage non-universel

Titre original : Elliptic solutions of generalized Brans-Dicke gravity with a non-universal coupling
Auteurs : Jean-Michel Alimi, Anastasia Golubtsova et Vincent Reverdy
Année : 2013
Journal : European Physical Journal C
arXiv : arXiv:1311.6384
Résumé : Nous étudions ici un modèle de gravité de Brans-Dicke généralisé présenté à la fois dans le référentiel de Jordan et dans le référentiel d'Einstein, ces derniers étant reliés par des transformations conformes. Nous montrons que les équations des champs scalaires dans le référentiel d'Einstein peuvent se réduire aux équations géodésiques de l'espace cible d'un sigma-model non-linéaire. Les solutions analytiques en termes de fonctions elliptiques sont obtenues pour des couplages conformes en exponentielles réciproques. Le comportement du facteur d'échelle est étudié numériquement. Pour certains ensembles de paramètres, des comportements accélérés sont obtenus. Une approximation analytique des solutions en fonction exponentielle est également donnée.

C.2 Oscillations acoustiques des baryons limitées par la variance cosmique au sein de la simulation DEUS-FUR ΛCDM

Titre original : Cosmic variance limited Baryon Acoustic Oscillations from the DEUS-FUR ΛCDM simulation
Auteurs : Yann Rasera, Pier-Stefano Corasaniti, Jean-Michel Alimi, Vincent Bouillot, Vincent Reverdy et Irène Balmès
Année : 2014
Journal : Monthly Notices of the Royal Astronomical Society
arXiv : arXiv:1311.5662
Résumé : Nous explorons l'évolution non-linéaire des oscillations acoustiques des baryons (BAO) dans le spectre de puissance de la matière à bas redshift de la simulation DEUS-FUR ΛCDM. Il s'agit de la première simulation à N-corps d'un volume équivalent à celui de l'Univers Observable, permettant ainsi d'accéder à des prédictions limitées par la variance cosmique à l'échelle des BAOs. Les effets des erreurs numériques systématiques ont été contrôlés en utilisant un ensemble de simulations haute résolution de grand volume. L'analyse combinée nous permet de mesurer le spectre de puissance entre $z = 0$ et $z = 1$ à 1% sur la plage d'échelle complète des BAOs, $0.03 < k\,[\mathrm{h/Mpc}] < 0.3$, pour des intervalles de taille $\delta k/k \lesssim 1\%$. Nous définissons les BAOs à partir d'un spectre non perturbé évolué non de manière non-linéaire et nous déterminons les caractéristiques des BAOs sans recourir à l'extrapolation de fonctions ajustées globalement. Nous quantifions les effets des non-linéarités sur la position et l'amplitude des extrema des BAOs, ainsi que le couplage à la composante générale du spectre de puissance. Nous utilisons ces estimateurs pour tester les prédictions non-linéaires de modèles semi-analytiques. À partir de l'analyse de l'évolution des BAOs avec le redshift nous trouvons que le second creux et le troisième pic restent insensibles aux effets non-linéaires. De plus, nous trouvons que le carré du facteur d'amortissement et du décalage de la position des extrema des BAOs varie en bonne approximation comme le carré du facteur de croissance, conformément à la théorie des perturbations. Cela confirme l'idée selon laquelle, qu'en plus de donner une mesure des distances cosmiques, une mesure précise des BAOs à différents redshifts peut fournir une mesure directe de la croissance des structures cosmiques.

C.3 Matrice de covariance du spectre de puissance de la matière dans les simulations DEUS-PUR ΛCDM : résolution en masse et erreurs non-gaussiennes

Titre original : Matter Power Spectrum Covariance Matrix from the DEUS-PUR ΛCDM simulations : Mass Resolution and non-Gaussian Errors
Auteurs : Linda Blot, Pier-Stefano Corasaniti, Jean-Michel Alimi, Vincent Reverdy et Yann Rasera
Année : 2014
Journal : Monthly Notices of the Royal Astronomical Society
arXiv : 1406.2713
Résumé : La prochaine génération des grands relevés de galaxies sondera la distribution de la matière dans l'Univers avec une précision inégalée. Les mesures du spectre de puissance de la matière à différentes échelles et à différents redshifts fourniront de fortes contraintes sur les paramètres cosmologiques. Cependant, sur des échelles non-linéaires, cela nécessitera une évaluation précise de la matrice de covariance. Dans cet article, nous calculons la matrice de covariance du spectre de puissance de la matière pour la cosmologie ΛCDM à partir d'un ensemble de simulations N-corps du projet Dark Energy Universe Simulation - Parallel Universe Runs (DEUS-PUR). Cet ensemble comprend 12288 simulations d'un volume de $(656\mathrm{Mpc/h})^3$ avec 256^3 particules. Nous combinons ces dernières avec un ensemble supplémentaire de 96 simulations de même volume mais avec 1024^3 particules chacune pour estimer l'effet des incertitudes non-gaussiennes dues aux effets de résolution. Nous trouvons que cela

est une source importante d'erreurs systématiques à haut redshift et à des échelles intermédiaires. Nous introduisons une méthode statistique empirique pour corriger cet effet et nous fournissons une détermination précise de la matrice de covariance sur une large gamme d'échelles incluant celle des oscillations acoustiques des baryons. Le grand ensemble statistique des simulations DEUS-PUR nous permet d'estimer finement la distribution de probabilité du spectre de puissance de la matière. Contrairement aux études précédentes basées sur des ensembles de simulations N-corps plus restreints, nous trouvons que le cœfficient de dissymétrie de la distribution s'écarte significativement des prédictions issues d'un champ de densité aléatoire gaussien à $k \gtrsim 0.25\,\mathrm{h/Mpc}$ et pour $z < 0.5$. Cela suggère que dans le cas de relevés de volume fini, un estimateur non-biaisé de la moyenne d'ensemble de la puissance à ces échelles et à ces redshifts pourrait nécessiter une meilleure prise en compte des erreurs non-gaussiennes que ce qui avait pu être considéré jusqu'à maintenant.

C.4 La première simulation à l'échelle de l'Univers Observable

Titre original : First-Ever Full Observable Universe Simulation
Auteurs : Jean-Michel Alimi, Vincent Bouillot, Yann Rasera, Vincent Reverdy, Pier-Stefano Corasaniti, Irène Balmès, Stéphane Requena, Xavier Delaruelle et Jean-Noël Richet
Année : 2012
Journal : Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis
arXiv : 1206.2838
Résumé : Nous avons réalisé la première simulation N-corps de la taille de l'Univers Observable (DEUS « Dark Energy Universe Simulation » FUR « Full Universe Run »). Cette simulation a suivi l'évolution de 550 milliards de particules sur une grille de raffinement adaptatif de plus de deux mille milliards de points de grilles sur toute l'histoire de l'Univers et sur 6 ordres de grandeur d'échelles, de la taille de la Voie Lactée jusqu'à celle de l'Univers Observable. À l'heure actuelle, il s'agit de la plus grande simulation cosmologique jamais réalisée. Elle fournit des informations uniques quant à la formation et à l'évolution des structures les plus massives de l'Univers et constitue un support exceptionnel aux programmes observationnels futurs dédiés à la cartographie de la distribution de matière et de galaxies dans l'Univers. La simulation a été réalisée sur 4752 nœuds fins (parmi 5040) du supercalculateur Curie, en utilisant plus de 300 Tio de mémoire pour un total de 10 millions d'heures de calcul. Environ 50 Pio de données ont été générées durant le calcul. En utilisant une chaîne de traitement avancée, il a été possible de réduire cette quantité à 500 Tio.

C.5 DEUS Full Observable Universe Simulations : défi numérique et perspectives

Titre original : DEUS Full Observable Universe Simulations : Numerical Challenge and Outlooks
Auteurs : Vincent Reverdy, Jean-Michel Alimi, Vincent Bouillot, Yann Rasera, Pier-Stefano Corasaniti, Irène Balmès, Stéphane Requena, Xavier Delaruelle et Jean-Noël Richet
Année : 2014
Journal : Advances in Engineering Software
Résumé : Nous rapportons la réalisation des premières simulations à l'échelle de l'Univers Observable. Ces simulations ont été réalisées sur 4752 nœuds du supercalculateur Curie au sein du projet DEUS-FUR (Dark Energy Universe Simulation : Full Universe Runs) dont le but est d'établir de nouvelles observables à même de contraindre la nature de l'énergie noire en étudiant la croissance des grandes structures, les caractéristiques des évènements extrêmes ou la distribution de la matière dans l'espace des redshifts. Le défi numérique de la première simulation DEUS-FUR correspondant au modèle de concordance ΛCDM a déjà été présenté à l'occasion de la conférence SuperComputing 2012. Ici, nous commençons par nous focaliser sur les aspects numériques liés à deux nouvelles simulations. Dans la

pratique, chacune de ces simulations a consisté à faire évoluer 550 milliards de particules dans une grille de raffinement adaptative, et l'une des nouvelles simulations a nécessité 2200 milliards de points de grille comparativement aux 2000 milliards du modèle ΛCDM du fait de la formation d'un plus grand nombre de structures. Nous mettons en lumière les optimisations et les réglages qui ont été effectués pour permettre la réalisation un tel ensemble de simulations et nous résumons ensuite quelques leçons importantes apprises lors de celle-ci en vue de préparer des projets de calcul exascale.

Elliptic solutions of generalized Brans-Dicke gravity with a non-universal coupling

J. -M. Alimi[1,a], **A. A. Golubtsova**[2,a,b] **and V. Reverdy**[3,a]

(a) Laboratoire de Univers et Théories (LUTh), Observatoire de Paris, Place Jules Janssen 5, 92190 Meudon, France

(b) Institute of Gravitation and Cosmology, Peoples' Friendship University of Russia, 6 Miklukho-Maklaya Str., Moscow 117198, Russia

Abstract

We study a model of the generalized Brans-Dicke gravity presented in both the Jordan and in the Einstein frames, which are conformally related. We show that the scalar field equations in the Einstein frame are reduced to the geodesics equations on the target space of the nonlinear sigma-model. The analytical solutions in elliptical functions are obtained when the conformal couplings are given by reciprocal exponential functions. The behavior of the scale factor in the Jordan frame is studied using numerical computations. For certain parameters the solutions can describe an accelerated expansion. We also derive an analytical approximation in exponential functions.

1 Introduction

Scalar fields play an enormous role in studies of gravity, understanding dynamics of the Universe and the physical nature of its dark sector. First, various unified models of field theories predict the existence of scalar partners to the tensor gravity of General Relativity. The simplest generalizations of the Einstein's theory of gravity, in which in addition to the metric the gravitation interaction is mediated by a scalar field, are those of scalar-tensor theories. Second, recent observational evidence [1]-[6] indicates that the Universe is presently dominated by a component dubbed dark energy. One of the approaches to account dark energy is to introduce the cosmological constant in the framework of general relativity. However a huge and still unexplained fine-tuning of the cosmological constant value [7] has not been understood yet. Another widespread interpretation of dark energy is that of quintessence, which is described by a scalar field minimally coupled to Einstein gravity rolling down some self-interaction potential [8, 9]. To take into account the region where the equation of state is less than $\omega = -1$, the model with a phantom scalar field (i.e. with a negative kinetic energy), an extension of the quintessence model, was suggested in [10]. Variety of works in scalar-tensor gravity are devoted to a search of an alternate explanation of dark energy [11]. An additional interest in scalar-tensor theories arises from

[1]jean-michel.alimi@obspm.fr
[2]siedhe@gmail.com
[3]vincent.reverdy@obspm.fr

OBSERVATOIRE DE PARIS

Mon. Not. R. Astron. Soc. **000**, 000–000 (0000) Printed 25 February 2014 (MN LATEX style file v2.2)

Cosmic variance limited Baryon Acoustic Oscillations from the DEUS-FUR ΛCDM simulation

Y. Rasera*, P.-S. Corasaniti, J.-M. Alimi, V. Bouillot, V. Reverdy, and I. Balmès
CNRS, Laboratoire Univers et Théories (LUTh), UMR 8102 CNRS, Observatoire de Paris, Université Paris Diderot- Paris 7 ; 5 Place Jules Janssen, 92190 Meudon, France

ABSTRACT

We investigate the non-linear evolution of Baryon Acoustic Oscillations (BAO) in the low-redshift matter power spectrum from the DEUS-FUR ΛCDM model simulation. This is the first cosmological N-body simulation encompassing the full observable cosmic volume, thus allowing cosmic variance limited predictions at BAO scales. We control the effect of numerical systematic errors using a series of large volume high-resolution simulations. The combined analysis allows us to measure the matter power spectrum between $z = 0$ and 1 to 1% over the entire BAO range, $0.03 < k$ [h Mpc^{-1}] < 0.3, in bins of size $\Delta k/k \lesssim 1\%$. We define the BAO with respect to a non-linearly evolved wiggle-free spectrum and determine the characteristics of the BAO without recurring to extrapolation from global fitting functions. We quantify the effects of non-linearities on the position and amplitude of the BAO extrema, and the coupling to the broadband slope of the power spectrum. We use these estimates to test non-linear predictions from semi-analytical models. Quite remarkably from the analysis of the redshift evolution of BAO we find that the second dip and third peak remains unaltered by non-linear effects. Furthermore, we find that the square of the damping factor and the shift of the position of BAO extrema scale to good approximation as the square of the growth factor, in agreement with expectations from perturbation theory. This confirms the idea that, besides cosmic distances, an accurate measurement of BAO at different redshifts can directly probe the growth of cosmic structures.

Key words: Cosmology; N-body Simulations; Large-Scale Structures;

1 INTRODUCTION

The propagation of primeval acoustic waves in the coupled photon-baryon plasma before recombination (Sakharov 1965; Silk 1968; Peebles & Yu 1970) generates a distinct pattern of temperature and polarization anisotropies in the Cosmic Microwave Background (CMB). These have been measured with unprecedented accuracy by observations of the Wilkinson Microwave Anisotropy Probe (WMAP) (Spergel et al. 2003, 2007; Komatsu et al. 2009) and more recently by the Planck satellite (Planck Collaboration XVI 2013).

A similar imprint is present in the late time distribution of large-scale structures in the form of an oscillatory pattern in the matter power spectrum, the so-called Baryon Acoustic Oscillations (BAO). These have been detected using measurements of two-point galaxy correlation function from the Sloan Digital Sky Survey (SDSS) (Eisenstein et al. 2005; Huetsi 2006) and the galaxy power spectrum from the 2-degree Field (2dF) survey (Cole et al. 2005).

CMB observations have accurately determined the distance travelled by acoustic waves at decoupling (i.e. the sound horizon). Thus, the detection of BAO in the galaxy distribution can be used as a standard ruler to estimate the angular diameter distance as well as the Hubble rate at redshifts probed by galaxy surveys (see e.g. Eisenstein, Hu & Tegmark 1998; Blake & Glazebrook 2003; Seo & Eisenstein 2003). A new generation of galaxy surveys has been designed to measure the BAO at different redshifts to few percent error from which it will be possible to infer stringent bounds on the cosmological parameters (see e.g. Ivezic et al. 2008; Amendola et al. 2012; Dawson et al. 2013). This demands for equally accurate cosmological model predictions.

In the linear regime of cosmic structure formation BAO appear as a series of damped oscillations superimposed to the broadband shape of the matter power spectrum. These can be computed to the desired level of ac-

* email:yann.rasera@obspm.fr

© 0000 RAS

Mon. Not. R. Astron. Soc. **000**, 000–000 (0000) Printed 12 June 2014 (MN LATEX style file v2.2)

Matter Power Spectrum Covariance Matrix from the DEUS-PUR ΛCDM simulations: Mass Resolution and non-Gaussian Errors

L. Blot[1⋆], P.S. Corasaniti[1], J.-M. Alimi[1,2], V. Reverdy[1], Y. Rasera[1]
[1] *Laboratoire Univers et Théories, UMR 8102 CNRS, Observatoire de Paris, Université Paris Diderot, 5 Place Jules Janssen, 92190 Meudon, France*
[2] *Institut d'Astrophysique, UMR 7095 CNRS, Université Pierre et Marie Curie, 98bis Blvd Arago, 75014 Paris, France*

12 June 2014

ABSTRACT

The upcoming generation of galaxy surveys will probe the distribution of matter in the universe with unprecedented accuracy. Measurements of the matter power spectrum at different scales and redshifts will provide stringent constraints on the cosmological parameters. However, on non-linear scales this will require an accurate evaluation of the covariance matrix. Here, we compute the covariance matrix of the matter power spectrum for the concordance ΛCDM cosmology from an ensemble of N-body simulations of the Dark Energy Universe Simulation - Parallel Universe Runs (DEUS-PUR). This consists of 12288 realizations of a $(656\,h^{-1}\,\mathrm{Mpc})^3$ simulation box with 256^3 particles. We combine this set with an auxiliary sample of 96 simulations of the same volume with 1024^3 particles to assess the impact of non-Gaussian uncertainties due to mass resolution effects. We find this to be an important source of systematic errors at high redshift and small intermediate scales. We introduce an empirical statistical method to correct for this effect and provide an accurate determination of the covariance matrix over a wide range of scales including the Baryon Acoustic Oscillations. The large statistical sample of DEUS-PUR simulations enables us to finely sample the probability distribution of the matter power spectrum. Contrary to previous studies which have used smaller N-body simulation ensembles, we find the skewness of the distribution to significantly deviate from expectations of a Gaussian random density field at $k \gtrsim 0.25\,h\,\mathrm{Mpc}^{-1}$ and $z < 0.5$. This suggests that in the case of finite volume surveys an unbiased estimate of the ensemble averaged band power at these scales and redshifts may require a careful assessment of non-Gaussian errors more than previously considered.

1 INTRODUCTION

Surveys of the large scale structures have been providing insightful data for more than a decade now. Observational projects such as the 2-degree Field Galaxy Redshift Survey (2dFGRS, Percival et al. 2001; Cole et al. 2005) and the Sloan Digital Sky Survey (SDSS, Tegmark et al. 2004) have yielded unprecedented measurements of the clustering of matter on the large scales. These observations have made possible the first detection of the Baryon Acoustic Oscillations (BAO) signal (Eisenstein et al. 2005; Percival et al. 2007) and provided constraints on model parameters that are complementary to those obtained from other standard cosmological probes. The success of these projects has opened the way to a new generation of survey programs.

In the years to come the Dark Energy Survey[1] (DES), the Large Synoptic Survey Telescope[2] (LSST) or the Euclid mission[3] will map the distribution of galaxies in larger cosmic volumes and with higher sensitivity. Through a variety of probes, these surveys aim to achieve a few percent error on the determination of several cosmological parameters. However, multiple challenges need to be addressed. From the theoretical point of view one of the most challenging aspects concerns the availability of robust theoretical predictions of the clustering of matter at small scales, which is crucial to correctly interpret the data and infer unbiased constraints on model parameters.

At small scales and late times the gravitational collapse becomes a highly non-linear process. Thus, model predictions cannot rely on standard linear perturbation theory and require solving the complex dynamics of matter collapse through numerical simulations. The need for accurate predictions of the clustering of matter over large interval of scales has driven up the demand for large volume high-resolution N-body simulations (Angulo et al. 2012; Kim et al. 2011; Alimi et al. 2012). As an example the Baryon Oscillation Spectroscopic Survey (BOSS, Dawson et al. 2013) has recently determined the cosmic distance scale to one percent accuracy from measurements of the BAO spectrum in the range of modes $0.01 < k[\,h\,\mathrm{Mpc}^{-1}] < 0.30$ (Anderson et al. 2013). Future surveys will push these measurements even further. Predicting the matter power spectrum in such a large interval of scales at a few percent level requires N-body simulations that cover very large cosmic volumes to reduce cosmic variance uncertainties at low k and with sufficient mass and spatial resolution to limit numerical systematic errors at large k. However, this is still not sufficient to correctly interpret the data since an unbiased statistical analysis also require knowledge of the covariance matrix.

If the initial matter density field is Gaussian distributed, dur-

[1] www.darkenergysurvey.org
[2] www.lsst.org
[3] www.euclid-ec.org

© 0000 RAS

First-Ever Full Observable Universe Simulation

Jean-Michel Alimi, Vincent Bouillot
Yann Rasera, Vincent Reverdy,
Pier-Stefano Corasaniti & Irène Balmès
LUTh, CNRS, Observatoire de Paris, Univ. Paris Diderot ;
5 Place Jules Janssen, 92190 Meudon, France.
http://www.deus-consortium.org

Stéphane Requena
GENCI
12, rue de l'Eglise,
75015 Paris, France.
http://www.genci.fr

Xavier Delaruelle
& Jean-Noel Richet
TGCC, CEA DAM, DIF
91297 Arpajon, France.
http://www-hpc.cea.fr

***Abstract*—We have performed the first-ever numerical N-body simulation of the full observable universe (DEUS "Dark Energy Universe Simulation" FUR "Full Universe Run"). This has evolved 550 billion particles on an Adaptive Mesh Refinement grid with more than two trillion computing points along the entire evolutionary history of the universe and across 6 order of magnitudes length scales, from the size of the Milky Way to that of the whole observable Universe. To date, this is the largest and most advanced cosmological simulation ever run. It provides unique information on the formation and evolution of the largest structure in the universe and an exceptional support to future observational programs dedicated to mapping the distribution of matter and galaxies in the universe. The simulation has run on 4752 (of 5040) thin nodes of BULL supercomputer CURIE, using more than 300 TB of memory for 10 million hours of computing time. About 50 PBytes of data were generated throughout the run. Using an advanced and innovative reduction workflow the amount of useful stored data has been reduced to 500 TBytes.**

I. Introduction

Over the past decades cosmological observations have provided us with a clearer picture of our universe [23], [29], [32], [34], [40]. On the one hand these measurements have confirmed the pillars of the Standard Hot Big-Bang scenario [28], on the other hand they showed the existence of a class of phenomena whose study has opened a window on a completely new and unexplored territory in physics. The atoms which makes planets, stars, the diffuse gas between galaxies and ourselves account only for a small fraction of the total content of the universe. Quite astonishingly 95% of the cosmos is in the form of two invisible components: 27% is in Cold Dark Matter (CDM) particles, which are primarily responsible for the formation of the visible structures in the universe [27]; 73% is in a unknown exotic form, dubbed "dark energy" (DE), which is responsible for the present phase of cosmic accelerated expansion [8].

Super-symmetric particles and extensions of the Standard Model of particles physics can provide physically motivated candidates to the role of Cold Dark Matter (CDM). In contrast, there is little clue on Dark Energy (DE). The presence of a cosmological constant term Λ in Einstein's equations of General Relativity [38] can account for phenomenon and the so-called "concordance" ΛCDM scenario has emerged as the minimal model to fit all available cosmological observations. However, the measured value of Λ is hardly reconcilable with any physical interpretation so far proposed. As a result, Dark Energy may well be of different origin. Several alternative scenarios have been advanced, but to date a coherent physical understanding of DE is still missing.

In the lack of a theoretical guidance, cosmologists are turning to observations and in the future several observational big-science projects such as BOSS [6], DES [12], LSST [26] and the EUCLID mission [15] will map with great accuracy the distribution of matter in the universe. These may eventually shed new light on the problem, as the clustering of matter may be key to infer the properties of DE. Indeed the physics responsible for DE may alter the time evolution of the clustering of CDM which in turn shapes the formation of the visible structures in the Universe.

What imprints does Dark Energy leave on the cosmic structures ? And inversely, how can the nature of Dark Energy be inferred from observations of the distribution of matter in the universe ?

These are the fundamental questions that the Dark Energy Universe Simulation [10] (DEUS) project seeks to answer.

The galaxy and clusters we observe today are the result of the growth of tiny density fluctuations present in the early universe, which we observe today as temperature fluctuations in the Cosmic Microwave Background (CMB) radiation. The gravitational infall of initial dark matter density fluctuations has evolved over time to reach a highly non-linear dynamical regime in which dark matter particles bound into stable objects, the halos. It is inside dark matter halos that cooling gas falls in to form stars and galaxies and it is the succession of halo merging processes that shapes the final distribution of large scale structures we observe today. Therefore, in order to study the imprint of Dark Energy on the cosmic structure formation one has to follow the gravitational collapse of Dark Matter throughout the history of the universe and across several order of magnitude length scales. This is not realizable using solely analytical methods, which break down as soon as non-linearity develops in the dynamics of dark matter. Only numerical N-body simulations provide the tool to follow the entire evolution of Dark Matter particles in an expanding universe dominated by Dark Energy.

During the past 10 years several groups have pushed to the limits both size and resolution of cosmological N-body simulations. The Millenium Simulation in 2005 has run a 2.2 billion light-years simulation box with 10 billion particles [35].

SC12, November 10-16, 2012, Salt Lake City, Utah, USA
978-1-4673-0806-9/12/$31.00 ©2012 IEEE

DEUS Full Observable Universe Simulations: Numerical Challenge and Outlooks

Vincent Reverdy[a], Jean-Michel Alimi[a], Vincent Bouillot[a], Yann Rasera[a], Pier-Stefano Corasaniti[a], Irène Balmès[a], Stéphane Requena[b], Xavier Delaruelle[c], Jean-Noel Richet[c]

[a] *LUTh, CNRS, Observatoire de Paris, Univ. Paris Diderot; 5 place Jules Janssen, 92190 Meudon, France*
[b] *GENCI, 12 rue de l'Eglise, 75015 Paris, France*
[c] *TGCC, CEA DAM, DIF, 91297 Arpajon, France*

Abstract

We report the realization of the first cosmological simulations on the scale of the whole Observable Universe. These simulations have been carried out on 4752 nodes of the Curie Supercomputer as a part of the DEUS-FUR (Dark Energy Universe Simulation: Full Universe Runs) project which aims at establishing new probes to put constaints on the nature of dark energy by comparing the growth of large scale structures, the characteristics of the extreme statistical events or the matter distribution in redshift space. The numerical challenge of the first DEUS-FUR simulation associated to the concordance ΛCDM model has already been presented during the 2012 Supercomputing conference[1]. Here we first focus on the numerical aspects of the two new simulations. In practice, each one of these simulations has evolved 550 billion dark matter particles in an Adaptive Mesh Refinement grid, and one of the new simulations has pushed back the total number of grid points from 2000 billion for the ΛCDM model to 2200 billion due to the formation of a larger number of structures. We highlight the optimizations and adjustments required to run such a set of simulations and we then summarize some important lessons learnt toward future exascale computing projects.

Keywords: Supercomputing, Cosmological simulation, N-body simulation, Dark Energy, Dark Matter, Exascale

1. Introduction

Since the formulation of the General Relativity in 1915 and the beginning of the scientific cosmology in the sense we know today, the continuous dialogue between theory and observations have led to a set of models to describe the Universe and its evolution as a physical object [2, 3, 4, 5, 6]. The "Big-Bang" model of an expanding Universe starting with a very hot and dense state has been formulated and refined throughout the second half of XXth century providing a theoretical framework in a good agreement with observational data. However, in 1998 the discovery that the luminosity of Type IA supernovae is lower than expected by two international teams has profoundly changed the cosmological paradigm [7, 8]. Such lower luminosity results from the acceleration of the expansion rate of the Universe in the last billions years which has been confirmed by several independent astronomical probes [9]. This unexpected discovery on Supernovae was rewarded by the Nobel Prize in Physics in 2011. It lies at the origin of one of the current biggest open problems in fundamental physics: *Why is expanding Universe accelerating?*

Numerous models of dark energy (real or effective energy component responsible for the cosmic acceleration) have been suggested [9]. In order to distinguish models within the theoretical interpretation set, the Dark Energy Universe Simulation project aims at shedding a new light on the following questions: *What imprints does dark energy leave on the cosmic structures? And inversely, how can the nature of dark energy be inferred from observations of the distribution of matter in the Universe?*

Answering these questions requires analyses of nonlinear gravitational collapse in an expanding Universe with dark energy. To do so, large numerical simulations are just the perfect tool: over the last decades, they have become a powerful mean of study for cosmology. From the first Jim Peebles' simulations in the 70's with 300 particles, cosmologists have regularly pushed the limits of the largest supercomputing power available at a given time to increase the size, the spatial resolution and the mass resolution of their simulations in order to improve both the statistical properties of their sample and the ac-

Bibliographie

***

[1] P. Fosalba, E. Gaztañaga, F. J. Castander, and M. Manera. The onion universe : all sky lightcone simulations in spherical shells. MNRAS, 391 :435–446, November 2008.

[2] R. Teyssier, S. Pires, S. Prunet, D. Aubert, C. Pichon, A. Amara, K. Benabed, S. Colombi, A. Refregier, and J.-L. Starck. Full-sky weak-lensing simulation with 70 billion particles. A&A, 497 :335–341, April 2009.

[3] Patrick Peter and Jean-Philippe Uzan. Primordial cosmology. Oxford Graduate Texts. Oxford Univ. Press, Oxford, 2009.

[4] P. J. E. Peebles. Principles of Physical Cosmology. 1993.

[5] S. Baeßler, B. R. Heckel, E. G. Adelberger, J. H. Gundlach, U. Schmidt, and H. E. Swanson. Improved Test of the Equivalence Principle for Gravitational Self-Energy. Physical Review Letters, 83 :3585–3588, November 1999.

[6] R. J. Nemiroff and B. Patla. Adventures in Friedmann cosmology : A detailed expansion of the cosmological Friedmann equations. American Journal of Physics, 76 :265–276, March 2008.

[7] Planck Collaboration, P. A. R. Ade, N. Aghanim, C. Armitage-Caplan, M. Arnaud, M. Ashdown, F. Atrio-Barandela, J. Aumont, C. Baccigalupi, A. J. Banday, and et al. Planck 2013 results. XVI. Cosmological parameters. ArXiv e-prints, March 2013.

[8] D. W. Hogg. Distance measures in cosmology. ArXiv Astrophysics e-prints, May 1999.

[9] I. M. H. Etherington. On the Definition of Distance in General Relativity. Philosophical Magazine, 15 :761, 1933.

[10] J. A. S. Lima, J. V. Cunha, and V. T. Zanchin. Deformed Distance Duality Relations and Supernova Dimming. ApJ, 742 :L26, December 2011.

[11] G. Bertone. Particle Dark Matter : Observations, Models and Searches. Cambridge University Press, 2010.

[12] D. Clowe, M. Bradač, A. H. Gonzalez, M. Markevitch, S. W. Randall, C. Jones, and D. Zaritsky. A Direct Empirical Proof of the Existence of Dark Matter. ApJ, 648 :L109–L113, September 2006.

[13] A. G. Riess, A. V. Filippenko, P. Challis, A. Clocchiatti, A. Diercks, P. M. Garnavich, R. L. Gilliland, C. J. Hogan, S. Jha, R. P. Kirshner, B. Leibundgut, M. M. Phillips, D. Reiss, B. P. Schmidt, R. A. Schommer, R. C. Smith, J. Spyromilio, C. Stubbs, N. B. Suntzeff, and J. Tonry. Observational Evidence from Supernovae for an Accelerating Universe and a Cosmological Constant. AJ, 116 :1009–1038, September 1998.

[14] S. Perlmutter, G. Aldering, G. Goldhaber, R. A. Knop, P. Nugent, P. G. Castro, S. Deustua, S. Fabbro, A. Goobar, D. E. Groom, I. M. Hook, A. G. Kim, M. Y. Kim, J. C. Lee, N. J. Nunes, R. Pain, C. R. Pennypacker, R. Quimby, C. Lidman, R. S. Ellis, M. Irwin, R. G. McMahon, P. Ruiz-Lapuente, N. Walton, B. Schaefer, B. J. Boyle, A. V. Filippenko, T. Matheson, A. S. Fruchter, N. Panagia, H. J. M. Newberg, W. J. Couch, and T. S. C. Project. Measurements of Ω and Λ from 42 High-Redshift Supernovae. ApJ, 517 :565–586, June 1999.

[15] T. Giannantonio, R. Scranton, R. G. Crittenden, R. C. Nichol, S. P. Boughn, A. D. Myers, and G. T. Richards. Combined analysis of the integrated Sachs-Wolfe effect and cosmological implications. Phys. Rev. D, 77(12) :123520, June 2008.

[16] S. Ho, C. Hirata, N. Padmanabhan, U. Seljak, and N. Bahcall. Correlation of CMB with large-scale structure. I. Integrated Sachs-Wolfe tomography and cosmological implications. Phys. Rev. D, 78(4) :043519, August 2008.

[17] T. S. van Albada, J. N. Bahcall, K. Begeman, and R. Sancisi. Distribution of dark matter in the spiral galaxy NGC 3198. ApJ, 295 :305–313, August 1985.

[18] E. Komatsu, K. M. Smith, J. Dunkley, C. L. Bennett, B. Gold, G. Hinshaw, N. Jarosik, D. Larson, M. R. Nolta, L. Page, D. N. Spergel, M. Halpern, R. S. Hill, A. Kogut, M. Limon, S. S. Meyer, N. Odegard, G. S. Tucker, J. L. Weiland, E. Wollack, and E. L. Wright. Seven-year Wilkinson Microwave Anisotropy Probe (WMAP) Observations : Cosmological Interpretation. ApJS, 192 :18, February 2011.

[19] Bicep2 Collaboration. Detection of B-Mode Polarization at Degree Angular Scales by BICEP2. Physical Review Letters, 112(24) :241101, June 2014.

[20] D. J. Mortlock, S. J. Warren, B. P. Venemans, M. Patel, P. C. Hewett, R. G. McMahon, C. Simpson, T. Theuns, E. A. Gonzáles-Solares, A. Adamson, S. Dye, N. C. Hambly, P. Hirst, M. J. Irwin, E. Kuiper, A. Lawrence, and H. J. A. Röttgering. A luminous quasar at a redshift of z = 7.085. Nature, 474 :616–619, June 2011.

[21] R. J. Bouwens, G. D. Illingworth, I. Labbe, P. A. Oesch, M. Trenti, C. M. Carollo, P. G. van Dokkum, M. Franx, M. Stiavelli, V. González, D. Magee, and L. Bradley. A candidate redshift z˜10 galaxy and rapid changes in that population at an age of 500Myr. Nature, 469 :504–507, January 2011.

[22] B. Famaey and S. S. McGaugh. Modified Newtonian Dynamics (MOND) : Observational Phenomenology and Relativistic Extensions. Living Reviews in Relativity, 15 :10, September 2012.

[23] I. Zlatev, L. Wang, and P. J. Steinhardt. Quintessence, Cosmic Coincidence, and the Cosmological Constant. Physical Review Letters, 82 :896–899, February 1999.

[24] T. Buchert. Dark Energy from structure : a status report. General Relativity and Gravitation, 40 :467–527, February 2008.

[25] R. R. Caldwell. A phantom menace ? Cosmological consequences of a dark energy component with super-negative equation of state. Physics Letters B, 545 :23–29, October 2002.

[26] R. R. Caldwell, M. Kamionkowski, and N. N. Weinberg. Phantom Energy : Dark Energy with w < -1 Causes a Cosmic Doomsday. Physical Review Letters, 91(7) :071301, August 2003.

[27] L. Wang, R. R. Caldwell, J. P. Ostriker, and P. J. Steinhardt. Cosmic Concordance and Quintessence. ApJ, 530 :17–35, February 2000.

[28] E. J. Copeland, M. Sami, and S. Tsujikawa. Dynamics of Dark Energy. International Journal of Modern Physics D, 15 :1753–1935, 2006.

[29] B. Ratra and P. J. E. Peebles. Cosmological consequences of a rolling homogeneous scalar field. Phys. Rev. D, 37 :3406–3427, June 1988.

[30] P. Brax and J. Martin. Robustness of quintessence. Phys. Rev. D, 61(10) :103502, May 2000.

[31] P. Binétruy. Models of dynamical supersymmetry breaking and quintessence. Phys. Rev. D, 60(6) :063502, September 1999.

[32] A. Masiero, M. Pietroni, and F. Rosati. SUSY QCD and quintessence. Phys. Rev. D, 61(2) :023504, January 2000.

[33] O. Lahav and A. R Liddle. The Cosmological Parameters 2014. ArXiv e-prints, January 2014.

[34] S. D. M. White, G. Efstathiou, and C. S. Frenk. The amplitude of mass fluctuations in the universe. MNRAS, 262 :1023–1028, June 1993.

[35] J.-M. Alimi, A. Füzfa, V. Boucher, Y. Rasera, J. Courtin, and P.-S. Corasaniti. Imprints of dark energy on cosmic structure formation - I. Realistic quintessence models and the non-linear matter power spectrum. MNRAS, 401 :775–790, January 2010.

[36] J.-M. Alimi, V. Bouillot, Y. Rasera, V. Reverdy, P.-S. Corasaniti, I. Balmes, S. Requena, X. Delaruelle, and J.-N. Richet. DEUS Full Observable ΛCDM Universe Simulation : the numerical challenge. ArXiv e-prints, June 2012.

[37] C. H. Brans. Mach's Principle and the Locally Measured Gravitational Constant in General Relativity. Physical Review, 125 :388–396, January 1962.

[38] R. H. Dicke. Cosmology, Mach's Principle and Relativity. American Journal of Physics, 31 :500–509, July 1963.

[39] T. Damour and K. Nordtvedt. General relativity as a cosmological attractor of tensor-scalar theories. Physical Review Letters, 70 :2217–2219, April 1993.

[40] A. Füzfa and J.-M. Alimi. Toward a unified description of dark energy and dark matter from the abnormally weighting energy hypothesis. Phys. Rev. D, 75(12) :123007, June 2007.

[41] T. Clifton, P. G. Ferreira, A. Padilla, and C. Skordis. Modified gravity and cosmology. Phys. Rep., 513 :1–189, March 2012.

[42] J.-M. Alimi and A. Füzfa. The abnormally weighting energy hypothesis : the missing link between dark matter and dark energy. J. Cosmology Astropart. Phys., 9 :14, September 2008.

[43] J.-M. Alimi, A. A. Golubtsova, and V. Reverdy. Elliptic solutions of generalized Brans-Dicke gravity with a non-universal coupling. ArXiv e-prints, November 2013.

[44] C. P. Ahn, R. Alexandroff, C. Allende Prieto, F. Anders, S. F. Anderson, T. Anderton, B. H. Andrews, É. Aubourg, S. Bailey, F. A. Bastien, and et al. The Tenth Data Release of the Sloan Digital Sky Survey : First Spectroscopic Data from the SDSS-III Apache Point Observatory Galactic Evolution Experiment. ApJS, 211 :17, April 2014.

[45] R. Laureijs, J. Amiaux, S. Arduini, J. . Augueres, J. Brinchmann, R. Cole, M. Cropper, C. Dabin, L. Duvet, A. Ealet, and et al. Euclid Definition Study Report. ArXiv e-prints, October 2011.

[46] I. Horvath, J. Hakkila, and Z. Bagoly. The largest structure of the Universe, defined by Gamma-Ray Bursts. ArXiv e-prints, November 2013.

[47] M. Colless, G. Dalton, S. Maddox, W. Sutherland, P. Norberg, S. Cole, J. Bland-Hawthorn, T. Bridges, R. Cannon, C. Collins, W. Couch, N. Cross, K. Deeley, R. De Propris, S. P. Driver, G. Efstathiou, R. S. Ellis, C. S. Frenk, K. Glazebrook, C. Jackson, O. Lahav, I. Lewis, S. Lumsden, D. Madgwick, J. A. Peacock, B. A. Peterson, I. Price, M. Seaborne, and K. Taylor. The 2dF Galaxy Redshift Survey : spectra and redshifts. MNRAS, 328 :1039–1063, December 2001.

[48] J. P. Huchra, L. M. Macri, K. L. Masters, T. H. Jarrett, P. Berlind, M. Calkins, A. C. Crook, R. Cutri, P. Erdoğdu, E. Falco, T. George, C. M. Hutcheson, O. Lahav, J. Mader, J. D. Mink, N. Martimbeau, S. Schneider, M. Skrutskie, S. Tokarz, and M. Westover. The 2MASS Redshift Survey Description and Data Release. ApJS, 199 :26, April 2012.

[49] P. Fouqué, J. M. Solanes, T. Sanchis, and C. Balkowski. Structure, mass and distance of the Virgo cluster from a Tolman-Bondi model. A&A, 375 :770–780, September 2001.

[50] H. M. Courtois, D. Pomarède, R. B. Tully, Y. Hoffman, and D. Courtois. Cosmography of the Local Universe. AJ, 146 :69, September 2013.

[51] G. de Vaucouleurs, A. de Vaucouleurs, H. G. Corwin, Jr., R. J. Buta, G. Paturel, and P. Fouqué. Third Reference Catalogue of Bright Galaxies. 1991.

[52] P. Schechter. An analytic expression for the luminosity function for galaxies. ApJ, 203 :297–306, January 1976.

[53] T. Buchert. Backreaction Issues in Relativistic Cosmology and the Dark Energy Debate. In M. Novello and S. E. Perez Bergliaffa, editors, XIIth Brazilian School of Cosmololy and Gravitation, volume 910 of American Institute of Physics Conference Series, pages 361–380, June 2007.

[54] S. Räsänen. Backreaction : directions of progress. Classical and Quantum Gravity, 28(16) :164008, August 2011.

[55] G. Hinshaw, M. R. Nolta, C. L. Bennett, R. Bean, O. Doré, M. R. Greason, M. Halpern, R. S. Hill, N. Jarosik, A. Kogut, E. Komatsu, M. Limon, N. Odegard, S. S. Meyer, L. Page, H. V. Peiris, D. N. Spergel, G. S. Tucker, L. Verde, J. L. Weiland, E. Wollack, and E. L. Wright. Three-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations : Temperature Analysis. ApJS, 170 :288–334, June 2007.

[56] P. J. E. Peebles. The large-scale structure of the universe. 1980.

[57] Y. B. Zel'dovich. Gravitational instability : An approximate theory for large density perturbations. A&A, 5 :84–89, March 1970.

[58] J. Hidding, S. F. Shandarin, and R. van de Weygaert. The Zel'dovich approximation : key to understanding cosmic web complexity. MNRAS, 437 :3442–3472, February 2014.

[59] W. J. Percival. Cosmological structure formation in a homogeneous dark energy background. A&A, 443 :819–830, December 2005.

[60] Yann Rasera. L'histoire cosmique des baryons dans un univers hierarchique. PhD thesis, Universite Paris-Diderot - Paris VII, 2005.

[61] H. Martel and P. R. Shapiro. A convenient set of comoving cosmological variables and their application. MNRAS, 297 :467–485, June 1998.

[62] J. F. Navarro, C. S. Frenk, and S. D. M. White. The Structure of Cold Dark Matter Halos. ApJ, 462 :563, May 1996.

[63] C. Lacey and S. Cole. Merger Rates in Hierarchical Models of Galaxy Formation - Part Two - Comparison with N-Body Simulations. MNRAS, 271 :676, December 1994.

[64] M. Davis, G. Efstathiou, C. S. Frenk, and S. D. M. White. The evolution of large-scale structure in a universe dominated by cold dark matter. ApJ, 292 :371–394, May 1985.

[65] E. Audit, R. Teyssier, and J.-M. Alimi. Non-linear dynamics and mass function of cosmic structures. II. Numerical results. A&A, 333 :779–789, May 1998.

[66] F. Roy, V. R. Bouillot, and Y. Rasera. pFoF : a highly scalable halo-finder for large cosmological data sets. A&A, 564 :A13, April 2014.

[67] A. Knebe, S. R. Knollmann, S. I. Muldrew, F. R. Pearce, M. A. Aragon-Calvo, Y. Ascasibar, P. S. Behroozi, D. Ceverino, S. Colombi, J. Diemand, K. Dolag, B. L. Falck, P. Fasel, J. Gardner, S. Gottlöber, C.-H. Hsu, F. Iannuzzi, A. Klypin, Z. Lukić, M. Maciejewski, C. McBride, M. C. Neyrinck, S. Planelles, D. Potter, V. Quilis, Y. Rasera, J. I. Read, P. M. Ricker, F. Roy, V. Springel, J. Stadel, G. Stinson, P. M. Sutter, V. Turchaninov, D. Tweed, G. Yepes, and M. Zemp. Haloes gone MAD : The Halo-Finder Comparison Project. MNRAS, 415 :2293–2318, August 2011.

[68] W. H. Press and P. Schechter. Formation of Galaxies and Clusters of Galaxies by Self-Similar Gravitational Condensation. ApJ, 187 :425–438, February 1974.

[69] R. K. Sheth and G. Tormen. Large-scale bias and the peak background split. MNRAS, 308 :119–126, September 1999.

[70] A. Jenkins, C. S. Frenk, S. D. M. White, J. M. Colberg, S. Cole, A. E. Evrard, H. M. P. Couchman, and N. Yoshida. The mass function of dark matter haloes. MNRAS, 321 :372–384, February 2001.

[71] J. Courtin, Y. Rasera, J.-M. Alimi, P.-S. Corasaniti, V. Boucher, and A. Füzfa. Imprints of dark energy on cosmic structure formation - II. Non-universality of the halo mass function. MNRAS, 410 :1911–1931, January 2011.

[72] I. Balmès, Y. Rasera, P.-S. Corasaniti, and J.-M. Alimi. Imprints of dark energy on cosmic structure formation - III. Sparsity of dark matter halo profiles. MNRAS, 437 :2328–2339, January 2014.

[73] Vincent Bouillot. Empreintes de l'Energie Noire sur la structuration de l'Univers. PhD thesis, Observatoire de Paris, 2012.

[74] K. S. Dawson and BOSS collaboration. The Baryon Oscillation Spectroscopic Survey of SDSS-III. AJ, 145 :10, January 2013.

[75] N. G. Busca and BOSS collaboration. Baryon acoustic oscillations in the Lyα forest of BOSS quasars. A&A, 552 :A96, April 2013.

[76] N. J. Mostek, C. Bebek, A. Dey, A. Ealet, S. Ho, J. Kneib, M. Lampton, M. Levi, K. Olsen, N. Padmanabhan, J. Pobar, E. Prieto, D. J. Schlegel, M. Sholl, A. Slozar, M. White, and BigBOSS Collaboration. BigBOSS : The Ground-Based Stage-IV BAO Experiment. In American Astronomical Society Meeting Abstracts 215, volume 42 of Bulletin of the American Astronomical Society, page 471.06, January 2010.

[77] K. Glazebrook. The WFMOS BAO survey. In Probing the Dark Universe with Subaru and Gemini, November 2005.

[78] R. E. Angulo, C. M. Baugh, C. S. Frenk, and C. G. Lacey. The detectability of baryonic acoustic oscillations in future galaxy surveys. MNRAS, 383 :755–776, January 2008.

[79] Y. Rasera, P.-S. Corasaniti, J.-M. Alimi, V. Bouillot, V. Reverdy, and I. Balmès. Cosmic-variance limited Baryon Acoustic Oscillations from the DEUS-FUR ΛCDM simulation. MNRAS, 440 :1420–1434, May 2014.

[80] D. J. Eisenstein and W. Hu. Baryonic Features in the Matter Transfer Function. ApJ, 496 :605–614, March 1998.

[81] W. J. Percival, R. C. Nichol, D. J. Eisenstein, D. H. Weinberg, M. Fukugita, A. C. Pope, D. P. Schneider, A. S. Szalay, M. S. Vogeley, I. Zehavi, N. A. Bahcall, J. Brinkmann, A. J. Connolly, J. Loveday, and A. Meiksin. Measuring the Matter Density Using Baryon Oscillations in the SDSS. ApJ, 657 :51–55, March 2007.

[82] H.-J. Seo and D. J. Eisenstein. Baryonic Acoustic Oscillations in Simulated Galaxy Redshift Surveys. ApJ, 633 :575–588, November 2005.

[83] M. Crocce and R. Scoccimarro. Nonlinear evolution of baryon acoustic oscillations. Phys. Rev. D, 77(2) :023533, January 2008.

[84] R. E. Smith, J. A. Peacock, A. Jenkins, S. D. M. White, C. S. Frenk, F. R. Pearce, P. A. Thomas, G. Efstathiou, and H. M. P. Couchman. Stable clustering, the halo model and non-linear cosmological power spectra. MNRAS, 341 :1311–1332, June 2003.

[85] J. Carlson, M. White, and N. Padmanabhan. Critical look at cosmological perturbation theory techniques. Phys. Rev. D, 80(4) :043531, August 2009.

[86] A. Taruya, F. Bernardeau, T. Nishimichi, and S. Codis. Direct and fast calculation of regularized cosmological power spectrum at two-loop order. Phys. Rev. D, 86(10) :103528, November 2012.

[87] J. Liske, A. Grazian, E. Vanzella, M. Dessauges, M. Viel, L. Pasquini, M. Haehnelt, S. Cristiani, F. Pepe, G. Avila, P. Bonifacio, F. Bouchy, H. Dekker, B. Delabre, S. D'Odorico, V. D'Odorico, S. Levshakov, C. Lovis, M. Mayor, P. Molaro, L. Moscardini, M. T. Murphy, D. Queloz, P. Shaver, S. Udry, T. Wiklind, and S. Zucker. Cosmic dynamics in the era of Extremely Large Telescopes. MNRAS, 386 :1192–1218, May 2008.

[88] C.-M. Yoo, T. Kai, and K.-I. Nakao. Redshift drift in Lemaître-Tolman-Bondi void universes. Phys. Rev. D, 83(4) :043527, February 2011.

[89] R. C. Tolman. Effect of Inhomogeneity on Cosmological Models. Proceedings of the National Academy of Science, 20 :169–176, March 1934.

[90] H. Bondi. Spherically symmetrical models in general relativity. MNRAS, 107 :410, 1947.

[91] P. Szekeres. A class of inhomogeneous cosmological models. Communications in Mathematical Physics, 41 :55–64, February 1975.

[92] T. Biswas and A. Notari. 'Swiss-cheese' inhomogeneous cosmology and the dark energy problem. J. Cosmology Astropart. Phys., 6 :21, June 2008.

[93] V. Marra, E. W. Kolb, S. Matarrese, and A. Riotto. Cosmological observables in a Swiss-cheese universe. Phys. Rev. D, 76(12) :123004, December 2007.

[94] P. Schneider. Gravitational lenses. 1999.

[95] M. Bartelmann and P. Schneider. Weak gravitational lensing. Phys. Rep., 340 :291–472, January 2001.

[96] C. M. Hirata and U. Seljak. Reconstruction of lensing from the cosmic microwave background polarization. Phys. Rev. D, 68(8) :083002, October 2003.

[97] D. J. Bacon and B. M. Schäfer. Twist and turn : weak lensing image distortions to second order. MNRAS, 396 :2167–2175, July 2009.

[98] A. Cooray. Integrated Sachs-Wolfe effect : Large scale structure correlation. Phys. Rev. D, 65(10) :103510, May 2002.

[99] S. Hilbert, J. Hartlap, S. D. M. White, and P. Schneider. Ray-tracing through the Millennium Simulation : Born corrections and lens-lens coupling in cosmic shear and galaxy-galaxy lensing. A&A, 499 :31–43, May 2009.

[100] M. Killedar, P. D. Lasky, G. F. Lewis, and C. J. Fluke. Gravitational lensing with three-dimensional ray tracing. MNRAS, 420 :155–169, February 2012.

[101] S. Räsänen. Applicability of the linearly perturbed FRW metric and Newtonian cosmology. Phys. Rev. D, 81(10) :103512, May 2010.

[102] V. F. Mukhanov, H. A. Feldman, and R. H. Brandenberger. Theory of cosmological perturbations. Phys. Rep., 215 :203–333, June 1992.

[103] N. E. Chisari and M. Zaldarriaga. Connection between Newtonian simulations and general relativity. Phys. Rev. D, 83(12) :123505, June 2011.

[104] R. K. Sachs. Gravitational waves in general relativity. vi. the outgoing radiation condition. Proc. R. Soc. London, Ser. A, 264 :309–338, 1961.

[105] C. Bonvin, R. Durrer, and M. A. Gasparini. Fluctuations of the luminosity distance. Phys. Rev. D, 73(2) :023523, January 2006.

[106] P. Fleury. Swiss-cheese models and the Dyer-Roeder approximation. J. Cosmology Astropart. Phys., 6 :54, June 2014.

[107] J.-P. Bruneton and J. Larena. Observables in a lattice Universe : the cosmological fitting problem. Classical and Quantum Gravity, 30(2) :025002, January 2013.

[108] George Marsaglia. Choosing a point from the surface of a sphere. The Annals of Mathematical Statistics, 43(2) :645–646, 04 1972.

[109] R. W. Hockney and J. W. Eastwood. Computer simulation using particles. 1988.

[110] C. J. Fluke, R. L. Webster, and D. J. Mortlock. The ray-bundle method for calculating weak magnification by gravitational lenses. MNRAS, 306 :567–574, July 1999.

[111] K. Yoshikawa, N. Yoshida, and M. Umemura. Direct Integration of the Collisionless Boltzmann Equation in Six-dimensional Phase Space : Self-gravitating Systems. ApJ, 762 :116, January 2013.

[112] E. I. Vorobyov, S. Recchi, and G. Hensler. Self-gravitating equilibrium models of dwarf galaxies and the minimum mass for star formation. A&A, 543 :A129, July 2012.

[113] TOP500 Supercomputer Site.

[114] Jack J. Dongarra, Piotr Luszczek, and Antoine Petitet. The linpack benchmark : Past, present, and future. concurrency and computation : Practice and experience. Concurrency and Computation : Practice and Experience, 15 :2003, 2003.

[115] V. Springel, S. D. M. White, A. Jenkins, C. S. Frenk, N. Yoshida, L. Gao, J. Navarro, R. Thacker, D. Croton, J. Helly, J. A. Peacock, S. Cole, P. Thomas, H. Couchman, A. Evrard, J. Colberg, and F. Pearce. Simulations of the formation, evolution and clustering of galaxies and quasars. Nature, 435 :629–636, June 2005.

[116] R. Teyssier. Cosmological hydrodynamics with adaptive mesh refinement. A new high resolution code called RAMSES. A&A, 385 :337–364, April 2002.

[117] V. Springel. E pur si muove : Galilean-invariant cosmological hydrodynamical simulations on a moving mesh. MNRAS, 401 :791–851, January 2010.

[118] O. Hahn, T. Abel, and R. Kaehler. A new approach to simulating collisionless dark matter fluids. MNRAS, 434 :1171–1191, September 2013.

[119] L. Blot, P. Stefano Corasaniti, J.-M. Alimi, V. Reverdy, and Y. Rasera. Matter Power Spectrum Covariance Matrix from the DEUS-PUR *Lambda*CDM simulations : Mass Resolution and non-Gaussian Errors. ArXiv e-prints, June 2014.

[120] S. Prunet, C. Pichon, D. Aubert, D. Pogosyan, R. Teyssier, and S. Gottloeber. Initial Conditions For Large Cosmological Simulations. ApJS, 178 :179–188, October 2008.

[121] T. Guillet and R. Teyssier. A simple multigrid scheme for solving the Poisson equation with arbitrary domain boundaries. Journal of Computational Physics, 230 :4756–4771, June 2011.

[122] Tom Goodale, Gabrielle Allen, Gerd Lanfermann, Joan Massó, Thomas Radke, Edward Seidel, and John Shalf. The Cactus framework and toolkit : Design and applications. In Vector and Parallel Processing – VECPAR'2002, 5th International Conference, Lecture Notes in Computer Science, Berlin, 2003. Springer.

[123] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http ://eigen.tuxfamily.org, 2010.

[124] D. Vandevoorde and N.M. Josuttis. C++ Templates : The Complete Guide. Pearson Education, 2002.

[125] ISO. International Organization for Standardization, Geneva, Switzerland, February.

[126] Andrew Sutton. N4040 : Working draft, c++ extensions for concepts, 2014.

[127] Matus Chochlik. N4111 : Static reflection (rev. 2), 2014.

[128] Richard Smith. N3652 : Relaxing constraints on constexpr functions constexpr member functions and implicit const, 2013.

[129] Wilhelm Ackermann. Zum hilbertschen aufbau der reellen zahlen. Mathematische Annalen, 99(1) :118–133, 1928.

[130] Andrei Alexandrescu. Modern C++ Design : Generic Programming and Design Patterns Applied. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[131] John J. Barton and Lee R. Nackman. Scientific and Engineering C++ : An Introduction with Advanced Techniques and Examples. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1994.

[132] Sanguthevar Rajasekaran and John Reif. Handbook of Parallel Computing : Models, Algorithms and Applications (Chapman & Hall/Crc Computer & Information Science Series). 1 edition.

[133] M. S. Warren and J. K. Salmon. A parallel hashed oct-tree n-body algorithm. In Proceedings of the 1993 ACM/IEEE Conference on Supercomputing, Supercomputing '93, pages 12–21, New York, NY, USA, 1993. ACM.

[134] Yedidya Hilewitz and Ruby B. Lee. Fast bit compression and expansion with parallel extract and parallel deposit instructions. pages 65–72, 11/09/2006 2006. (Best Paper Award).

[135] Yedidya Hilewitz and Ruby B. Lee. Fast bit gather, bit scatter and bit permutation instructions for commodity microprocessors. Journal of Signal Processing Systems, 53 :145–169, 11/2008 2008.

[136] DAVID R. MUSSER. Introspective sorting and selection algorithms. Software : Practice and Experience, 27(8) :983–993, 1997.

[137] The Boost Graph Library : User Guide and Reference Manual. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.

[138] D. Gao, Y. Zhang, and Y. Zhao. The Application of kd-tree in Astronomy. In R. W. Argyle, P. S. Bunclark, and J. R. Lewis, editors, Astronomical Data Analysis Software and Systems XVII, volume 394 of Astronomical Society of the Pacific Conference Series, page 525, August 2008.

[139] Bengt Fornberg. Numerical differentiation of analytic functions. ACM Trans. Math. Softw., 7(4) :512–526, December 1981.

[140] Abraham. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. Analytical Chemistry, 36(8) :1627–1639, 1964.

[141] Makoto Matsumoto and Takuji Nishimura. Mersenne twister : A 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Trans. Model. Comput. Simul., 8(1) :3–30, January 1998.

[142] M. Berger. Géométrie vivante ou L'échelle de Jacob. Nouvelle bibliothèque mathématique. Cassini, 2009.

[143] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. Numerical Recipes 3rd Edition : The Art of Scientific Computing. Cambridge University Press, New York, NY, USA, 3 edition, 2007.

[144] P. W. Anderson. Science, (4047) :393–396.

[145] I. Y. Aref'eva, L. V. Joukovskaya, and S. Y. Vernov. Bouncing and accelerating solutions in nonlocal stringy models. Journal of High Energy Physics, 7 :87, July 2007.

[146] W. Romberg. Vereinfachte numerische Integration. Det Kongelige Norske Videnskabers Selskab Forhandlinger, 1955.

[147] Accurate computation of f'(x) and f'(x) f.

[148] R W D Nickalls. Viète, descartes and the cubic equation. The Mathematical Gazette, 90(518) :203–208, July 2006.

[149] C. W. Misner, K. S. Thorne, and J. A. Wheeler. Gravitation. 1973.

[150] E. T. Whittaker and G. N. Watson. A Course of Modern Analysis. Cambridge University Press, fourth edition, 1927. Reprinted 1990.

OBSERVATOIRE DE PARIS