

Méthode Numérique pour les équations de Landau-Lifshitz-Gilbert



Étudiante :

Marie De Penguern

Responsables du stage :

M. Nicolas Crouseilles

M. Adrien Laurent

Remerciements

Je souhaite premièrement remercier le directeur de l'IRMAR M. Mihai Gradinaru et le directeur M. Patrick Gros de m'avoir accueilli dans leur établissement.

Je voudrais aussi remercier mon maître de stage M. Nicolas Crouseilles de m'avoir permis de travailler dans son équipe et de m'avoir accompagnée sur toute la durée de ce stage.

Je tiens de plus à remercier M. Adrien Laurent pour le temps qu'il m'a accordée. Il a eu la patience et la bonté de répondre à toutes mes questions de manière pédagogique et a su éveiller ma curiosité tout au long du stage.

Finalement je désire remercier M. Antoine Guines pour m'avoir permis de trouver ce stage qui m'aura fait progresser.

Table des matières

| | | |
|----------|---|----------|
| 1 | Présentation de l'entreprise | 5 |
| 1.1 | INRIA | 5 |
| 1.2 | L'équipe MINGuS | 5 |
| 2 | Développement du problème traité | 6 |
| 2.1 | Introduction | 6 |
| 2.2 | Propriétés | 7 |
| 2.3 | Schémas | 8 |
| 2.3.1 | Rappels et définitions des schémas numériques | 9 |
| 2.3.2 | Méthode d'Euler explicite | 10 |
| 2.3.3 | Méthode d'Euler implicite | 11 |
| 2.3.4 | Méthode d'Euler exponentielle | 11 |
| 2.3.5 | Méthode de splitting | 12 |
| 2.4 | Test des méthodes | 19 |
| 2.4.1 | Sans amortissement | 19 |
| 2.4.2 | Avec amortissement | 22 |
| 2.5 | Conclusion | 25 |

Résumé

J'ai effectué dans mon stage à l'INRIA dans l'équipe MINGuS dont les bureaux sont situés l'IRMAR (Institut de recherche en mathématiques) situé à quelque pas de l'INRIA. Le groupe MINGuS étudie les équations aux dérivées partielles, y compris les versions stochastiques, en lien avec la physique des plasmas et quantique, avec un accent sur les phénomènes multi-échelles.

Le sujet de mon stage concernait l'approximation numérique de l'équation de Landau-Lifshitz-Gilbert pour la modélisation du spin ionique.

Le spin ionique correspond au moment magnétique intrinsèque des ions dans un matériau, influençant ses propriétés magnétiques et électroniques. Les équations de Landau-Lifshitz-Gilbert (LLG) modélisent la dynamique de ce spin, en décrivant comment il réagit aux champs magnétiques et aux interactions internes. Ces équations sont cruciales pour comprendre et prévoir le comportement des dispositifs magnétiques à l'échelle nanométrique, ce qui impacte le développement de technologies telles que les mémoires magnétiques et les dispositifs spintroniques.

L'équipe avec laquelle je travaillais avait une collaboration avec des physiciens à Strasbourg avec un code de simulation en cours de développement à ce sujet. L'objectif de mon stage était donc de vérifier et compléter l'étude déjà commencée.

Le travail à effectuer concernait tout d'abord une bibliographie des documents déjà existants au sujet des équations de Land-Lifshitz-Gilbert des propriétés et des schémas numériques déjà testés sur ces équations.

Une fois imprégnée du sujet et des contraintes à respecter pour fournir une simulation convenable, j'ai du moi-même développer plusieurs méthodes numériques afin de trouver celle qui correspondrait le mieux à cette étude. J'ai pu tester des méthodes numériques étudiées à l'INSA dans le cours d'équation aux dérivées partielles ainsi que la méthode de décomposition en éléments finis. Cependant j'ai pu aussi m'initier à de nouvelles méthodes numériques plus complexes.

J'ai du ensuite programmer ces méthodes et effectuer des tests avec différents paramètres pour exploiter les points forts et les faiblesses de chacune des méthodes pour finalement déterminer celle qui convenait le mieux.

Les équations de LLG possèdent un terme d'amortissement. L'étude a ainsi donc débuté par des tests sans perturbation pour vérifier la validité du model puis nous avons ajouter l'amortissement.

1 Présentation de l'entreprise

J'ai effectué mon stage de spécialité à l'INRIA de Rennes dans l'équipe MINGuS. Mon stage était d'une durée de 10 semaines que j'ai coupé en deux en raison de ma mobilité en Italie. J'ai d'abord fait 7 semaines en janvier, puis 3 semaines fin août.

1.1 INRIA

Inria, l'institut national français de recherche en sciences et technologies du numérique, est depuis janvier 2024 à la tête de l'Agence de programmes dans le numérique, conçue pour dynamiser la coopération entre l'enseignement supérieur et la recherche. L'Inria se distingue par son engagement envers une recherche de calibre mondial, l'innovation technologique, et la prise de risques entrepreneuriale. Répartis au sein de 220 équipes-projets, dont beaucoup sont en partenariat avec de grandes universités, plus de 3 800 chercheurs explorent de nouvelles perspectives, souvent en collaboration interdisciplinaire avec des acteurs industriels, pour relever des défis ambitieux. En tant qu'institut technologique, Inria soutient un large éventail d'innovations, allant de l'édition de logiciels open source à la création de startups technologiques (Deeptech).

1.2 L'équipe MINGuS

J'ai plus particulièrement fait mon stage à l'INRIA de Rennes qui elle compte 30 équipes de recherches pour un total de 600 chercheurs. J'ai travaillé dans l'équipe MINGuS (Multiscale Numerical Geometric Schemes) spécialisée dans les Schémas géométriques numériques à plusieurs échelles. Le groupe de recherche MINGuS se concentre sur l'analyse des équations aux dérivées partielles, y compris les versions stochastiques, en lien avec la physique des plasmas et quantique. Il étudie particulièrement les phénomènes multi-échelles, comme les comportements oscillatoires et dissipatifs, en utilisant des modèles tels que les équations de Schrödinger et de Vlasov, pour des applications en dynamique des gaz raréfiés, plasmas, et physique quantique. Cette équipe est spécialement en partenariat avec l'École normale supérieure de Rennes, le CNRS, et l'Université de Rennes. J'ai plus précisément travaillé avec le chef d'équipe Nicolas Crouseilles et Adrien Laurent un chercheur de l'équipe.

Néanmoins, j'ai effectué mon stage dans le bâtiment de l'IRMAR car mes responsables de stage y font partie du pôle Analyse.

2 Développement du problème traité

2.1 Introduction

Le spin des ions est une propriété fondamentale en physique quantique, qui joue un rôle crucial dans de nombreux phénomènes physiques, notamment dans les domaines de la magnétisme et de la spintronique. Le spin peut être considéré comme une sorte de moment angulaire intrinsèque associé à une particule, et il est souvent représenté par un vecteur qui pointe dans une direction spécifique dans l'espace.

L'aspect physique du spin des ions est profondément lié aux équations de Landau-Lifshitz-Gilbert (LLG), qui décrivent l'évolution temporelle des aimants et des spins dans un matériau magnétique. Ces équations jouent un rôle central dans la modélisation et la compréhension des phénomènes de magnétisme, en tenant compte des interactions entre les spins, la structure cristalline du matériau et les champs magnétiques appliqués.

Il existe 2 formes de solutions [Rug16] :

Forme de Landau-Lifshitz :

$$\partial_t m = -\frac{1}{1+\alpha} m \times h_{eff} - \frac{\alpha}{1+\alpha^2} m \times (m \times h_{eff})$$

Forme de Gilbert :

$$\partial_t m = -m \times h_{eff} + \alpha m \times \partial_t m$$

Avec ici, m l'aimantation unitaire, h_{eff} le champ magnétique effectif et α le coefficient d'amortissement. Les deux équations décrivent la même dynamique physique mais sous des formes mathématiques différentes. La première équation est souvent utilisée dans les simulations numériques car elle sépare clairement les contributions de la précession et de l'amortissement, ce qui peut simplifier certaines approches numériques.

De plus, $h_{eff} = \frac{\partial H}{\partial m_i}$ avec $H = \text{hamiltonien}$, $m_i = \text{spin en } x = x_i$, avec $x_{i+1} - x_i = \alpha$.

$$H = -\sum_{i=0}^N \sum_{j \in v(i)} m_i \cdot m_j \text{ avec } v(i) = \text{voisinage du site } i = \{i-1, i+1\}$$

L'équation de LL posée sur $x = x_i$, $i = 0, \dots, N$ s'écrit :

$$\partial_t m_i = -m_i \times h_{eff} = -m_i \times \left(-\sum_{j \in v(i)} m_j \right) = m_i \times (m_{i+1} + m_{i-1}) \quad (1)$$

On souhaite obtenir une équation posée sur $x \in [0, L]$. Pour cela on effectue un développement de Taylor.

$$m_{i \pm 1} = m(x_{i \pm 1}) = m(x_i) \pm \alpha (\partial_x m)(x_i) + \frac{\alpha^2}{2} (\partial_x^2 m)(x_i) + O(\alpha^3)$$

et l'équation (1) devient (en remplaçant m_i par $m(x_i)$) :

$$\begin{aligned} \partial_t m(x_i) &= +m(x_i) \times (2m(x_i) + \alpha^2 (\partial_x^2 m)(x_i)) + O(\alpha^3) \\ &= \alpha^2 m(x_i) \times (\partial_x^2 m)(x_i) + O(\alpha^3) \end{aligned}$$

et on étend cette équation posée en $x = x_i$ à tout $x \in [0, L]$:

$$\partial_t m = \alpha^2 m \times \partial_x^2 m$$

Soit $m(t, x) \in \mathbf{S}^2$, $t \geq 0$, $x \in [0, \frac{2\pi}{k}]$

On se propose donc dans un premier temps d'étudier l'équation suivante :

$$\boxed{\partial_t m = am \times \partial_x^2 m + bm \times (m \times \partial_x^2 m)}$$

Avec $m(0, x) = \begin{pmatrix} \frac{\epsilon \cos(kx)}{\sqrt{1+\epsilon^2}} \\ \frac{\epsilon \sin(kx)}{\sqrt{1+\epsilon^2}} \\ 1 \end{pmatrix}$ k fixé car on souhaite que $|m| = 1$ [Bañ04]. Nous nous placerons de plus dans les conditions de Dirichlet afin de simplifier l'étude.

2.2 Propriétés

Norme

En effectuant le produit scalaire par m de l'équation, on obtient :

$$\partial_t m \cdot m = \frac{1}{2} \partial_t |m|^2 = 0$$

Ce qui implique la conservation de la magnitude de l'alimentation et donc $|m(t)| = |m(0)| = 1$. C'est la condition que nous allons chercher à respecter tout au long de nos simulations.

Relation de dispersion

On souhaite de plus trouver une solution sous la forme :

$$m(t, x) = m_0 + \epsilon m_1(t, x)$$

afin de modéliser des petites perturbations sur la solution et avec m_0 un vecteur constant. Par exemple $m_0 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

Ainsi, c'est l'expression de $m_1(t, x)$ que nous allons chercher. En remplaçant dans l'équation et en ne conservant que les termes d'ordre ϵ , on obtient :

$$\partial_t m_1 = a m_0 \times \partial_x^2 m_1 + b m_0 \times (m_0 \times \partial_x^2 m_1)$$

Ce qui nous donne :

$$\partial_t \begin{pmatrix} m_{1,1} \\ m_{1,2} \\ m_{1,3} \end{pmatrix} = a \begin{pmatrix} -\partial_x^2 m_{1,2} \\ \partial_x^2 m_{1,1} \\ 0 \end{pmatrix} + b \begin{pmatrix} -\partial_x^2 m_{1,1} \\ -\partial_x^2 m_{1,2} \\ 0 \end{pmatrix} \quad (2)$$

Pour mener à bien l'analyse linéaire, il est d'usage d'effectuer des transformées de Fourier en espace. On définit la série de Fourier de m :

$$m(x) = \sum_{k \in \mathbb{Z}} \hat{m}_k e^{ikx} \quad (3)$$

où \hat{m}_k sont les coefficients de Fourier de m . Pour obtenir les coefficients de Fourier de $\partial_x m$, on dérive (3) par rapport à la variable x

$$\partial_x m = \sum_{k \in \mathbb{Z}} \hat{m}_k \partial_x (e^{ikx}) = \sum_{k \in \mathbb{Z}} (\hat{m}_k ik) e^{ikx},$$

et on déduit par identification que les coefficients de Fourier de $\partial_x m$ sont $(\hat{m}_k ik)$. De même, en dérivant 2 fois (3), on obtient que $(-k^2) \hat{m}_k$ sont les coefficients de $\partial_x^2 m$.

Ainsi, si on insère la transformée de Fourier à l'équation linéarisée (2), on obtient (voir [Son16] pour une étude similaire) :

$$\partial_t \begin{pmatrix} \hat{m}_{k,1,1} \\ \hat{m}_{k,1,2} \\ \hat{m}_{k,1,3} \end{pmatrix} = -ak^2 \begin{pmatrix} -\hat{m}_{k,1,2} \\ \hat{m}_{k,1,1} \\ 0 \end{pmatrix} + bk^2 \begin{pmatrix} \hat{m}_{k,1,1} \\ \hat{m}_{k,1,2} \\ 0 \end{pmatrix}$$

On applique enfin la transformée de Laplace en temps qui transforme $\partial_t \hat{m}$ en $i\omega \tilde{\hat{m}}$ (où la notation $\tilde{}$ représente les coefficients de la transformée de Laplace), ce qui permet d'obtenir (en enlevant les $\tilde{}$, $\hat{}$ et les indices k)

$$i\omega \begin{pmatrix} m_{1,1} \\ m_{1,2} \end{pmatrix} = -ak^2 \begin{pmatrix} -m_{1,2} \\ m_{1,1} \end{pmatrix} + bk^2 \begin{pmatrix} m_{1,1} \\ m_{1,2} \end{pmatrix}$$

On peut réécrire cette équation sous forme d'un système linéaire. On a :

$$\begin{pmatrix} bk^2 - i\omega & -ak^2 \\ ak^2 & bk^2 - i\omega \end{pmatrix} \begin{pmatrix} m_{1,1} \\ m_{1,2} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

et comme nous cherchons des perturbations $(m_{1,1}, m_{1,2})$ non nulles, on cherche une relation dite de dispersion qui annule le déterminant de la matrice.

En calculant le déterminant on obtient : $(bk^2 - i\omega)^2 + a^2k^4 = 0$ soit $bk^2 - i\omega = iak^2$ et donc $\omega = ak^2 - ibk^2$. C'est cette condition que l'on cherchera à vérifier dans les résultats numériques. La relation de dispersion fournit une relation directe entre les fréquences ω et les nombres d'onde k , ce qui signifie que pour chaque k , la fréquence correspondante ω est déterminée.

Quand on applique la transformée de Fourier à la solution $m(t, x)$ de l'équation LLG, on obtient un spectre en k et ω . Ce spectre permet de visualiser les composantes d'onde présentes dans la solution. Si on observe un pic dans ce spectre, il correspond à une onde particulière avec un certain nombre d'onde k et une fréquence ω . La relation de dispersion prédit précisément où ces pics devraient apparaître.

2.3 Schémas

Maintenant, afin de trouver une expression pour $m_1(t, x)$ dans le cas où il n'y a pas de terme d'amortissement, c'est à dire :

$$\partial_t m = am \times \partial_x^2 m$$

on souhaite avoir une expression sous la forme

$$\partial_t M = B(M)M$$

avec $B(M) \in \mathbb{R}^{3N_x \times 3N_x}$ et $M \in \mathbb{R}^{3N_x}$ afin de pouvoir appliquer nos schémas numériques.

En discrétisant, on a $m_i^n \approx m(n\Delta t, i\Delta x) \forall t \geq 0, x \in [0, \frac{2\pi}{k}]$, $n = 0, \dots, N_t$, $N_t\Delta t = T$, $\Delta t > 0$, $i = 0, \dots, N_x$, $\Delta x > 0$ et $N_x\Delta x = L$

$$\text{En notant } M = \begin{pmatrix} M^1 \\ M^2 \\ M^3 \end{pmatrix} \text{ où } M^1 = \begin{pmatrix} m^1(\Delta x) \\ m^1(2\Delta x) \\ \vdots \\ m^1(N_x\Delta x) \end{pmatrix} \text{ et donc } M = \begin{pmatrix} m^1(\Delta x) \\ m^1(2\Delta x) \\ \vdots \\ m^1(N_x\Delta x) \\ m^2(\Delta x) \\ \vdots \\ m^2(N_x\Delta x) \\ m^3(\Delta x) \\ \vdots \\ m^3(N_x\Delta x) \end{pmatrix}$$

On a

$$\Delta M = \frac{1}{\Delta x^2} \begin{pmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & A \end{pmatrix} M$$

Definition : Erreur locale de troncature

On appelle erreur locale de troncature ϵ_n définie par :

$$\epsilon_n = M(t_{n+1}) - M_{n+1}$$

Definition : Erreur globale de convergence

On appelle erreur globale de convergence e_n définie par :

$$e_n = M(T) - M^{Nt}$$

Definition : Ordre de convergence

La méthode à 1 pas est dite d'ordre $p \in \mathbb{N}$ si l'erreur locale satisfait $\exists h^* > 0$ tel que $\epsilon = O(h_n^{p+1})$, $h_n \leq h^*$. La méthode est consistante si et seulement si $p \geq 1$.

2.3.2 Méthode d'Euler explicite**Formulation**

Pour une équation différentielle du type

$$M' = aB(M)M + bD(M)M,$$

la méthode d'Euler explicite est donnée par :

$$M_{n+1} = M_n + \Delta t (aB(M_n)M_n + bD(M_n)M_n).$$

Preuve de l'ordre

Développement de Taylor de $M(t_{n+1})$:

$$\begin{aligned} M(t_{n+1}) &= M(t_n + \Delta t) \\ &= M(t_n) + \Delta t \frac{dM}{dt}(t_n) + \frac{\Delta t^2}{2} \frac{d^2M}{dt^2}(t_n) + O(\Delta t^3). \end{aligned}$$

En utilisant $\frac{dM}{dt} = aB(M)M + bD(M)M$, nous avons :

$$\begin{aligned} \frac{d^2M}{dt^2} &= a \frac{dB(M)}{dt} M + aB(M) \frac{dM}{dt} \\ &\quad + b \frac{dD(M)}{dt} M + bD(M) \frac{dM}{dt}. \end{aligned}$$

Alors :

$$\begin{aligned} M(t_{n+1}) &= M(t_n) + \Delta t (aB(M_n)M_n + bD(M_n)M_n) \\ &\quad + \frac{\Delta t^2}{2} (aB(M) (aB(M_n)M_n + bD(M_n)M_n) + bD(M) (aB(M_n)M_n + bD(M_n)M_n) + \text{autres termes}) \end{aligned}$$

Expression de l'erreur locale de troncature :

$$\begin{aligned} \epsilon_n &= M(t_{n+1}) - (M_n + \Delta t (aB(M_n)M_n + bD(M_n)M_n + O(\Delta t^2))) \\ &= O(\Delta t^2). \end{aligned}$$

Erreur globale :

$$E_n = O(n\Delta t^2) = O(\Delta t).$$

Donc, la méthode d'Euler explicite est d'ordre 1.

Propriétés géométriques

Non-conservation des propriétés géométriques : La méthode d'Euler explicite ne conserve pas les propriétés géométriques comme la norme de m ici que nous voulons maintenir à 1, ce qui peut mener à des erreurs accumulatives importantes pour les intégrations à long terme. [Hai+06]

2.3.3 Méthode d'Euler implicite

Formulation

La méthode d'Euler implicite est donnée par :

$$M_{n+1} = M_n + \Delta t (aB(M_{n+1})M_{n+1} + bD(M_{n+1})M_{n+1}).$$

Preuve de l'ordre

Développement de Taylor de $M(t_{n+1})$:

$$M(t_{n+1}) = M(t_n) + \Delta t \frac{dM}{dt}(t_n) + \frac{\Delta t^2}{2} \frac{d^2M}{dt^2}(t_n) + O(\Delta t^3).$$

En utilisant $\frac{dM}{dt} = aB(M)M + bD(M)M$, nous avons :

$$\begin{aligned} M(t_{n+1}) &= M(t_n) + \Delta t (aB(M_n)M_n + bD(M_n)M_n) \\ &\quad + \frac{\Delta t^2}{2} (aB(M) (aB(M_n)M_n + bD(M_n)M_n) + bD(M) (aB(M_n)M_n + bD(M_n)M_n) + \text{autres termes}) \end{aligned}$$

Expression de l'erreur locale de troncature :

$$\begin{aligned} \epsilon_n &= M(t_{n+1}) - (M_n + \Delta t (aB(M_{n+1})M_{n+1} + bD(M_{n+1})M_{n+1})) \\ &= \left(M(t_n) + \Delta t \frac{dM}{dt}(t_n) + \frac{\Delta t^2}{2} \frac{d^2M}{dt^2}(t_n) + O(\Delta t^3) \right) - (M_n + \Delta t (aB(M_{n+1})M_{n+1} + bD(M_{n+1})M_{n+1})) \\ &= O(\Delta t^2). \end{aligned}$$

Erreur globale :

$$E_n = O(n\Delta t^2) = O(\Delta t).$$

Donc, la méthode d'Euler implicite est également d'ordre 1.

Propriétés géométriques

Non-conservation des propriétés géométriques : Comme la méthode explicite, elle ne conserve pas les propriétés géométriques .

2.3.4 Méthode d'Euler exponentielle

Formulation de la méthode d'Euler exponentielle

Pour l'équation différentielle

$$M' = aB(M)M + bD(M)M,$$

la méthode d'Euler exponentielle est donnée par :

$$M_{n+1} = \exp(\Delta t (aB(M_n) + bD(M_n))) M_n.$$

Développement de Taylor de $M(t_{n+1})$

Développons $M(t_{n+1})$ en série de Taylor autour de t_n :

$$M(t_{n+1}) = M(t_n) + \Delta t \frac{dM}{dt}(t_n) + \frac{\Delta t^2}{2} \frac{d^2M}{dt^2}(t_n) + O(\Delta t^3).$$

Approximation de la méthode d'Euler exponentielle

Pour de petits Δt , nous pouvons utiliser l'approximation de l'exponentielle de matrice :

$$\exp(\Delta t (aB(M_n) + bD(M_n))) \approx I + \Delta t (aB(M_n) + bD(M_n)) + O(\Delta t^2).$$

Ainsi en reprenant la démonstration pour Euler explicite, on obtient que la méthode d'Euler exponentielle est d'ordre 1.

Propriétés géométriques

1. Conservation des intégrales premières : Certaines versions des méthodes exponentielles peuvent être conçues pour conserver des quantités spécifiques, comme l'énergie, sur de longues périodes de temps.
2. Adaptation aux systèmes linéaires : Elles sont particulièrement efficaces pour les systèmes linéaires ou linéarisables, où la solution peut être exprimée en termes d'exponentielles de matrices.
3. Complexité computationnelle : Elles peuvent être plus complexes à mettre en œuvre et nécessitent souvent le calcul d'exponentielles de matrices, ce qui peut être coûteux en termes de calcul. Dans notre cas nous travaillons avec des matrices de taille $3Nx$ ce qui est d'autant plus coûteux.

[Hai+06]

2.3.5 Méthode de splitting

Décomposition de l'équation

L'équation différentielle initiale

$$m'(t) = f(t, m(t))$$

est décomposée en plusieurs termes f_j , tels que

$$f(t, m(t)) = f_1(t, m(t)) + f_2(t, m(t)) + \dots + f_m(t, m(t)).$$

Cette décomposition peut se faire de différentes manières en fonction des propriétés spécifiques de l'équation originale [BCM24].

Résolution des sous-problèmes

Chaque sous-problème de la forme

$$m' = f_j(m), \quad m(0) = m_0,$$

est résolu séparément. Ces sous-problèmes sont choisis de manière à être plus faciles à intégrer, soit exactement, soit numériquement tout en préservant certaines propriétés structurelles (par exemple, symplectique, unitaire, etc.).

Combinaison des solutions

Les solutions des sous-problèmes sont ensuite combinées pour obtenir une approximation de la solution du problème original. Cette combinaison se fait de manière à préserver les propriétés qualitatives et géométriques de la solution originale de l'équation différentielle.

Preuve de l'ordre

Considérons l'équation différentielle initiale :

$$\frac{dM}{dt} = f(M),$$

où $f(M)$ est décomposée en trois parties : $f(M) = f_A(M) + f_B(M) + f_C(M)$.

Développement de Taylor de la solution exacte

Développons la solution exacte $M(t_{n+1})$ en série de Taylor autour de t_n :

$$M(t_{n+1}) = M(t_n) + \Delta t \frac{dM}{dt}(t_n) + \frac{\Delta t^2}{2} \frac{d^2M}{dt^2}(t_n) + O(\Delta t^3).$$

En utilisant $\frac{dM}{dt} = f(M)$, nous obtenons :

$$M(t_{n+1}) = M(t_n) + \Delta t f(M_n) + \frac{\Delta t^2}{2} f'(M_n) f(M_n) + O(\Delta t^3).$$

Approximation de la méthode de splitting

Pour la méthode de splitting, nous avons trois flux : ϕ^A , ϕ^B , et ϕ^C .

Étape 1 : Application de ϕ^A :

$$\phi^A(\Delta t, M^n) = M^n + \Delta t f_A(M^n) + \frac{\Delta t^2}{2} f'_A(M^n) f_A(M^n) + O(\Delta t^3).$$

Étape 2 : Application de ϕ^B :

$$\begin{aligned} \phi^B(\Delta t, \phi^A(\Delta t, M^n)) &= M^n + \Delta t f_A(M^n) + \Delta t f_B(M^n) \\ &+ \frac{\Delta t^2}{2} f'_A(M^n) f_A(M^n) + O(\Delta t^3). \end{aligned}$$

Étape 3 : Application de ϕ^C :

$$\begin{aligned} \phi^C(\Delta t, \phi^B(\Delta t, \phi^A(\Delta t, M^n))) &= M^n + \Delta t (f_A(M^n) + f_B(M^n) + f_C(M^n)) \\ &+ \frac{\Delta t^2}{2} f'_A(M^n) f_A(M^n) + O(\Delta t^3). \end{aligned}$$

Comparaison avec la solution exacte

La solution exacte après un pas de temps est :

$$M(t_{n+1}) = M(t_n) + \Delta t f(M_n) + \frac{\Delta t^2}{2} f'(M_n) f(M_n) + O(\Delta t^3),$$

où $f(M_n) = f_A(M_n) + f_B(M_n) + f_C(M_n)$.

La solution de la méthode de splitting est :

$$\begin{aligned} M^{n+1} &= M^n + \Delta t (f_A(M^n) + f_B(M^n) + f_C(M^n)) \\ &+ \frac{\Delta t^2}{2} f'_A(M^n) f_A(M^n) + O(\Delta t^3). \end{aligned}$$

Erreur locale de troncation

$$\begin{aligned} \epsilon_n &= M(t_{n+1}) - M^{n+1} \\ &= \left(M_n + \Delta t f(M_n) + \frac{\Delta t^2}{2} f'(M_n) f(M_n) + O(\Delta t^3) \right) \\ &\quad - \left(M_n + \Delta t (f_A(M^n) + f_B(M^n) + f_C(M^n)) + \frac{\Delta t^2}{2} f'_A(M^n) f_A(M^n) + O(\Delta t^3) \right) \\ &= \frac{\Delta t^2}{2} (f'(M_n) f(M_n) - f'_A(M^n) f_A(M^n)) + O(\Delta t^3). \end{aligned}$$

Puisque $f'(M_n) f(M_n)$ est une combinaison linéaire des termes $f'_A(M_n) f_A(M_n)$, $f'_B(M_n) f_B(M_n)$, et $f'_C(M_n) f_C(M_n)$, l'erreur locale de troncation est de l'ordre de $O(\Delta t^2)$.

Erreur globale

L'erreur globale après n étapes est donc :

$$E_n = O(n\Delta t^2) = O(\Delta t).$$

Ainsi, la méthode de splitting est d'ordre 1.

Propriétés géométriques

1. Conservation des intégrales premières : Les méthodes de splitting peuvent conserver des intégrales premières (comme l'énergie ou le moment angulaire) lorsqu'elles sont appliquées à des systèmes Hamiltoniens ou possédant d'autres symétries conservatrices.
2. Symplecticité : Certaines méthodes de splitting, comme les méthodes de Strang ou les méthodes symplectiques, préservent la structure symplectique des systèmes Hamiltoniens. Cela signifie qu'elles conservent la géométrie de l'espace de phase, ce qui est crucial pour obtenir des solutions numériques qui restent physiquement significatives sur de longues périodes de temps.
3. Réversibilité : Les méthodes de splitting peuvent être conçues pour être réversibles, ce qui signifie que la méthode d'intégration donne les mêmes résultats si le temps est inversé. Ceci est particulièrement utile pour les systèmes réversibles où les trajectoires doivent pouvoir être suivies dans les deux directions temporelles.
4. Préservation du volume : Pour certains systèmes, les méthodes de splitting peuvent être conçues pour préserver le volume dans l'espace de phase, ce qui est une propriété géométrique importante pour certains systèmes dynamiques.[Hai+06]

Formulation

En rappelant qu'avec $J = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ on a :

$$\exp(Jt) = \begin{pmatrix} \cos(t) & \sin(t) \\ -\sin(t) & \cos(t) \end{pmatrix}$$

et qu'une équation du type $\partial_t u = Au$, ($u(0) = u_0$) admet pour solution $u(t) = \exp(At)u_0$

On cherche à décomposer notre équation initiale de sorte à obtenir des matrices dont nous connaissons la valeur de l'exponentielle.

Ainsi, dans le cas **sans amortissement** :

$$\begin{aligned} \partial_t m &= am \times \partial_x^2 m \\ &= a \begin{pmatrix} 0 & \partial_x^2 m_3 & -\partial_x^2 m_2 \\ -\partial_x^2 m_3 & 0 & \partial_x^2 m_1 \\ \partial_x^2 m_2 & -\partial_x^2 m_1 & 0 \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} \\ &= \partial_x^2 m_1 \begin{pmatrix} 0 & 0 & 0 \\ 0 & J & 0 \\ 0 & 0 & 0 \end{pmatrix} m + \partial_x^2 m_3 \begin{pmatrix} J & 0 \\ 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} m + \partial_x^2 m_2 \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} m \end{aligned}$$

Ainsi, $m'(t) = f_v(t, m(t)) + f_r(t, m(t)) + f_b(t, m(t))$

Dans le cas **avec amortissement**, nous avons essayé un premier splitting.

En rappelant la formule :

$$\partial_t m = am \times \Delta m + bm \times (m \times \Delta m)$$

On remarque que

$$\partial_t m = a \begin{pmatrix} \boxed{m_2 \partial_x^2 m_3} & \boxed{-m_3 \partial_x^2 m_2} \\ \boxed{m_3 \partial_x^2 m_1} & \boxed{-m_1 \partial_x^2 m_3} \\ \boxed{m_1 \partial_x^2 m_2} & \boxed{-m_2 \partial_x^2 m_1} \end{pmatrix} + b \begin{pmatrix} \boxed{m_2 m_1 \partial_x^2 m_2} & \boxed{-m_2^2 \partial_x^2 m_1} & \boxed{-m_3^2 \partial_x^2 m_1} & \boxed{+m_1 m_3 \partial_x^2 m_3} \\ \boxed{m_3 m_2 \partial_x^2 m_3} & \boxed{-m_3^2 \partial_x^2 m_2} & \boxed{-m_1^2 \partial_x^2 m_2} & \boxed{+m_1 m_2 \partial_x^2 m_1} \\ \boxed{m_1 m_3 \partial_x^2 m_1} & \boxed{-m_1^2 \partial_x^2 m_3} & \boxed{-m_2^2 \partial_x^2 m_3} & \boxed{+m_2 m_3 \partial_x^2 m_2} \end{pmatrix}$$

On peut écrire le système **vert** de la manière suivante :

$$\begin{cases} \partial_t m_1 = am_2 \partial_x^2 m_3 + bm_1 m_3 \partial_x^2 m_3 \\ \partial_t m_2 = -am_1 \partial_x^2 m_3 + bm_2 m_3 \partial_x^2 m_3 \\ \partial_t m_3 = -b(m_1^2 + m_2^2) \partial_x^2 m_3 \end{cases}$$

Or comme nous sommes sur la sphère de rayon 1, on a :

$$m_1^2 + m_2^2 + m_3^2 = 1$$

On peut ainsi réécrire la dernière équation comme $\partial_t m_3 = -b(1 - m_3^2)m_3''$ On peut ainsi réécrire :

$$\partial_t m = \begin{pmatrix} 0 & -a\partial_x^2 m_3 & 0 \\ a\partial_x^2 m_3 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} m + \begin{pmatrix} -bm_1 m_3 \partial_x^2 m_3 \\ -bm_2 m_3 \partial_x^2 m_3 \\ b(1 - m_3^2) \partial_x^2 m_3 \end{pmatrix} = f_v(m)$$

Or $\langle m, f_v(m) \rangle = \langle m, \partial_t m \rangle = -b(m_1^2 + m_2^2)m_3 \partial_x^2 m_3 + b(m_1^2 + m_2^2)m_3 \partial_x^2 m_3 = 0$ ce qui nous permet d'affirmer que l'on préserve bien la norme.

On va ainsi calculer M_3 à chaque étape à l'aide d'un schéma exponentiel. On a la formule suivante :

$$M_3^{n+1} = \exp(-b\Delta t \begin{pmatrix} \sqrt{1 - (M_3^n)^2} & & \\ & \ddots & \\ & & \sqrt{1 - (M_3^n)^2} \end{pmatrix}) AM_3^n$$

On peut réécrire le système de la manière suivante :

$$M^{n+1} = f_{v,\Delta t}(M_n)$$

avec

$$f_{v,\Delta t}(M_n) = GF \begin{pmatrix} M^1 \\ M^2 \\ (M^3)^{n+1} \end{pmatrix}$$

$$G = \begin{pmatrix} \frac{\sqrt{1-M_3^{n+1}}}{\sqrt{1-M_3^n}} & & 0 & & 0 \\ & \ddots & & & \\ & & \frac{\sqrt{1-M_3^{n+1}}}{\sqrt{1-M_3^n}} & & 0 \\ 0 & & & \frac{\sqrt{1-M_3^{n+1}}}{\sqrt{1-M_3^n}} & 0 \\ & \ddots & & & \\ & & 0 & & \frac{\sqrt{1-M_3^{n+1}}}{\sqrt{1-M_3^n}} \\ 0 & & & & 1 \\ & \ddots & & & \\ & & 0 & & 0 & 1 \end{pmatrix}$$

$$F = \begin{pmatrix} \cos(a\Delta t A_1 M^3) & & \sin(a\Delta t A_1 M^3) & & 0 \\ & \ddots & & & \\ & & \cos(a\Delta t A_{N_x} M^3) & & \sin(a\Delta t A_{N_x} M^3) & 0 \\ -\sin(a\Delta t A_1 M^3) & & & \cos(a\Delta t A_1 M^3) & & 0 \\ & \ddots & & & & \\ & & -\sin(a\Delta t A_{N_x} M^3) & & \cos(a\Delta t A_1 M^3) & 0 \\ 0 & & & & & 1 \\ & \ddots & & & & \\ & & 0 & & & 0 & 1 \end{pmatrix}$$

On peut aussi écrire le système **rouge** de la manière suivante :

$$\partial_t m = \begin{pmatrix} 0 & 0 & -a\partial_x^2 m_2 \\ 0 & 0 & 0 \\ a\partial_x^2 m_2 & 0 & 0 \end{pmatrix} m + \begin{pmatrix} bm_2 m_1 \partial_x^2 m_2 \\ -b(1 - m_2^2) \partial_x^2 m_2 \\ bm_2 m_3 \partial_x^2 m_2 \end{pmatrix}$$

On va ainsi calculer M_2 à chaque étape à l'aide d'un schéma exponentiel comme on l'a fait pour M_3 dans le cas précédent. On a la formule suivante :

$$M_2^{n+1} = \exp(-b\Delta t \begin{pmatrix} \sqrt{1 - (M_2^n)^2} & & \\ & \ddots & \\ & & \sqrt{1 - (M_2^n)^2} \end{pmatrix}) AM_2^n$$

On peut réécrire le système de la manière suivante :

$$M^{n+1} = f_{r,\Delta t}(M^n)$$

avec

$$f_{r,\Delta t}(M_n) = GF \begin{pmatrix} M^1 \\ (M^2)^{n+1} \\ M^3 \end{pmatrix}$$

$$G = \begin{pmatrix} \frac{\sqrt{1-M_2^{n+1}}}{\sqrt{1-M_2^n}} & 0 & 0 & & & \\ & \ddots & & \ddots & & \ddots \\ & & \frac{\sqrt{1-M_2^{n+1}}}{\sqrt{1-M_2^n}} & 0 & & 0 \\ 0 & & 0 & 1 & 0 & \\ & \ddots & & \ddots & & \ddots \\ & & 0 & 0 & 1 & \\ 0 & & 0 & 0 & \frac{\sqrt{1-M_2^{n+1}}}{\sqrt{1-M_2^n}} & \\ & \ddots & & \ddots & & \ddots \\ & & 0 & 0 & 0 & \frac{\sqrt{1-M_2^{n+1}}}{\sqrt{1-M_2^n}} \end{pmatrix}$$

$$F = \begin{pmatrix} \cos(a\Delta t A_1 M^2) & & 0 & & -\sin(a\Delta t A_1 M^2) & & \\ & \ddots & & \ddots & & \ddots & \\ & & \cos(a\Delta t A_{N_x} M^2) & & 0 & & -\sin(a\Delta t A_{N_x} M^2) \\ 0 & & 0 & 1 & 0 & & \\ & \ddots & & \ddots & & \ddots & \\ & & 0 & 0 & 1 & & 0 \\ \sin(a\Delta t A_1 M^2) & & 0 & 0 & \cos(a\Delta t A_1 M^2) & & \\ & \ddots & & \ddots & & \ddots & \\ & & \sin(a\Delta t A_{N_x} M^2) & & 0 & & \cos(a\Delta t A_{N_x} M^2) \end{pmatrix}$$

Finalment, on peut aussi écrire le système **bleu** de la manière suivante :

$$\partial_t m = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -a\partial_x^2 m_1 \\ 0 & -a\partial_x^2 m_1 & 0 \end{pmatrix} m + \begin{pmatrix} -b(1 - m_1^2) \partial_x^2 m_1 \\ bm_1 m_2 \partial_x^2 m_1 \\ bm_1 m_3 \partial_x^2 m_1 \end{pmatrix}$$

On va ainsi calculer M_1 à chaque étape à l'aide d'un schéma exponentiel comme on l'a fait pour M_2 et M_3 dans le cas précédent. On a la formule suivante :

$$m_1^{n+1} = \exp(-b\Delta t \begin{pmatrix} \sqrt{1 - (M_1^n)^2} & & \\ & \ddots & \\ & & \sqrt{1 - (M_1^n)^2} \end{pmatrix}) AM_1^n$$

On peut réécrire le système de la manière suivante :

$$M^{n+1} = f_{b,\Delta t}^C(M^n)$$

avec

$$f_{b,\Delta t}(M_n) = GF \begin{pmatrix} (M^1)^{n+1} \\ M^2 \\ M^3 \end{pmatrix}$$

$$G = \begin{pmatrix} 1 & & 0 & & 0 & & 0 \\ & \ddots & & \ddots & & \ddots & \\ & & 1 & & 0 & & 0 \\ 0 & & & \frac{\sqrt{1-M_1^{n+1}}}{\sqrt{1-M_1^n}} & & 0 & \\ & \ddots & & & \ddots & \ddots & \\ & & 0 & & \frac{\sqrt{1-M_1^{n+1}}}{\sqrt{1-M_1^n}} & & 0 \\ 0 & & & & & \frac{\sqrt{1-M_1^{n+1}}}{\sqrt{1-M_1^n}} & \\ & \ddots & & & & & \ddots \\ & & 0 & & 0 & & \frac{\sqrt{1-M_1^{n+1}}}{\sqrt{1-M_1^n}} \end{pmatrix}$$

$$F = \begin{pmatrix} 1 & & 0 & & 0 & & 0 \\ & \ddots & & \ddots & & \ddots & \\ & & 1 & & 0 & & 0 \\ 0 & & \cos(a\Delta t A_1 M^1) & & \sin(a\Delta t A_1 M^1) & & \\ & \ddots & & \ddots & & \ddots & \\ & & 0 & & \cos(a\Delta t A_1 M^1) & & \sin(a\Delta t A_{N_x} M^1) \\ 0 & & -\sin(a\Delta t A_1 M^1) & & \cos(a\Delta t A_1 M^1) & & \\ & \ddots & & \ddots & & \ddots & \\ & & 0 & & -\sin(a\Delta t A_{N_x} M^1) & & \cos(a\Delta t A_{N_x} M^1) \end{pmatrix}$$

Ainsi, notre méthode de splitting peut s'écrire :

$$m^{n+1} = \phi^C(\phi^B(\phi^A(m^n)))$$

Splitting 2 On propose une deuxième forme de splitting :

$$m' = a \begin{pmatrix} \boxed{m_2 \partial_x^2 m_3} & \boxed{-m_3 \partial_x^2 m_2} \\ \boxed{m_3 \partial_x^2 m_1} & \boxed{-m_1 \partial_x^2 m_3} \\ \boxed{m_1 \partial_x^2 m_2} & \boxed{-m_2 \partial_x^2 m_1} \end{pmatrix} + b \begin{pmatrix} \boxed{m_2 m_1 \partial_x^2 m_2 - m_2^2 \partial_x^2 m_1} & \boxed{-m_3^2 \partial_x^2 m_1 + m_1 m_3 \partial_x^2 m_3} \\ \boxed{m_3 m_2 \partial_x^2 m_3 - m_3^2 \partial_x^2 m_2} & \boxed{-m_1^2 \partial_x^2 m_2 + m_1 m_2 \partial_x^2 m_1} \\ \boxed{m_1 m_3 \partial_x^2 m_1 - m_1^2 \partial_x^2 m_3} & \boxed{-m_2^2 \partial_x^2 m_3 + m_2 m_3 \partial_x^2 m_2} \end{pmatrix}$$

Ce qui nous donne pour la partie **verte** :

$$m' = \begin{pmatrix} 0 & a\partial_x^2 m_3 & 0 \\ -a\partial_x^2 m_3 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} m + b \begin{pmatrix} (m_1\partial_x^2 m_2 - m_2\partial_x^2 m_1)m_2 \\ -(m_1\partial_x^2 m_2 - m_2\partial_x^2 m_1)m_1 \\ 0 \end{pmatrix}$$

Ce qui peut se réécrire :

$$m' = \begin{pmatrix} 0 & a\partial_x^2 m_3 + b(m_1\partial_x^2 m_2 - m_2\partial_x^2 m_1) & 0 \\ -a\partial_x^2 m_3 - b(m_1\partial_x^2 m_2 - m_2\partial_x^2 m_1) & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} m$$

Pour la partie **rouge** :

$$m' = \begin{pmatrix} 0 & 0 & -a\partial_x^2 m_2 \\ 0 & 0 & 0 \\ a\partial_x^2 m_2 & 0 & 0 \end{pmatrix} m + b \begin{pmatrix} (m_3\partial_x^2 m_1 - m_1\partial_x^2 m_3)m_3 \\ 0 \\ -(m_1\partial_x^2 m_3 - m_3\partial_x^2 m_1)m_1 \end{pmatrix}$$

Ce qui peut se réécrire :

$$m' = \begin{pmatrix} 0 & 0 & a\partial_x^2 m_2 + b(m_3\partial_x^2 m_1 - m_1\partial_x^2 m_3) \\ 0 & 0 & 0 \\ a\partial_x^2 m_2 - b(m_1\partial_x^2 m_3 - m_3\partial_x^2 m_1) & 0 & 0 \end{pmatrix} m$$

Pour la partie **bleu** :

$$m' = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -a\partial_x^2 m_1 & 0 \end{pmatrix} + b \begin{pmatrix} 0 \\ -(m_3\partial_x^2 m_2 - m_2\partial_x^2 m_3)m_3 \\ (m_3\partial_x^2 m_2 - m_2\partial_x^2 m_3)m_2 \end{pmatrix}$$

Ce qui peut se réécrire :

$$m' = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & a\partial_x^2 m_1 - b(m_3\partial_x^2 m_2 - m_2\partial_x^2 m_3) \\ 0 & -a\partial_x^2 m_1 + b(m_3\partial_x^2 m_2 - m_2\partial_x^2 m_3) & 0 \end{pmatrix} m$$

Ainsi pour chaque partie on a

$$m' = Hm$$

avec H une matrice de taille $3Nx \times 3Nx$ et donc on a à chaque étape $M^{n+1} = \exp(dtH)M^n$.

2.4 Test des méthodes

Dans un premier temps, nous avons choisi de faire des test pour l'équation de Landau-Lifshitz, c'est à dire l'équation sans amortissement afin d'avoir une première idée des méthodes qui pourront être utilisées par la suite.

2.4.1 Sans amortissement

Euler implicite

Pour la méthode d'Euler implicite nous n'avons effectués des test que sur la forme sans amortissement car celle-ci se montrait déjà défailante.

Considérons les paramètres suivants : $N_t = 2000$, $\epsilon = 0.001$, $a = 1$, $N_x = 10$. Avec les paramètres

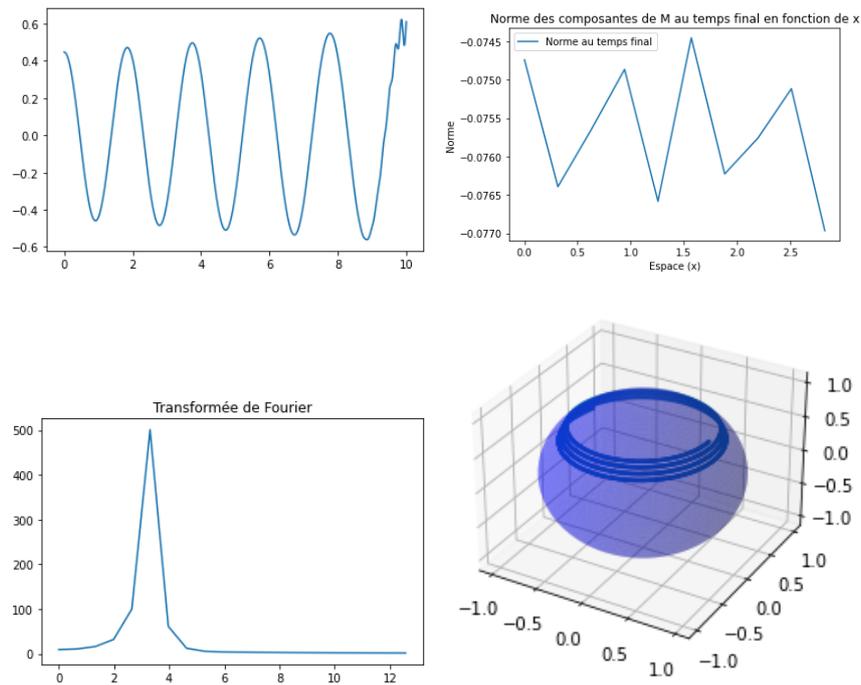


FIGURE 2 – Composante 1 en fonction du temps et Norme de $M - 1$ au temps N_t pour Euler implicite $k = 2$, $\epsilon = 0.001$, $a = 1$, $b = 0$, $N_x = 10$, $N_t = 2000$

précisé plus haut, $K = 2$, $T = 10$, $a = -1$, $b = 0$ et $\epsilon = 0.001$ et en rappelant la relation de dispersion : $\omega = ak^2 + ibk^2$ on calcul manuellement ω et on obtient :

$$\omega = (-1)2^2$$

$$\omega = -4$$

Ainsi, la partie réelle de ω vaut 4 ce qui correspond à la fréquence pour laquelle on obtient un pic sur la transformée de Fourier.

On reconnaît l'allure du cos pour la première composante mais son oscillation augmente et on voit une défailance après un certain temps. On remarque bien l'imprécision de la méthode ainsi qu'une différence trop grande de la norme de m avec la valeur 1 qui est négative. On exclura donc après la méthode de Euler implicite pour les tests suivants.

Euler explicite

On observe une meilleure régularité pour Euler explicite même si comme pour Euler implicite on aperçoit une explosion au temps final. Ainsi, il est nécessaire de choisir un pas de temps plus petit. La transformée de Fourier nous confirme la valeur de ω . Finalement la norme est mieux préservée que pour Euler implicite puisque la différence entre la norme de M et 1 est de l'ordre

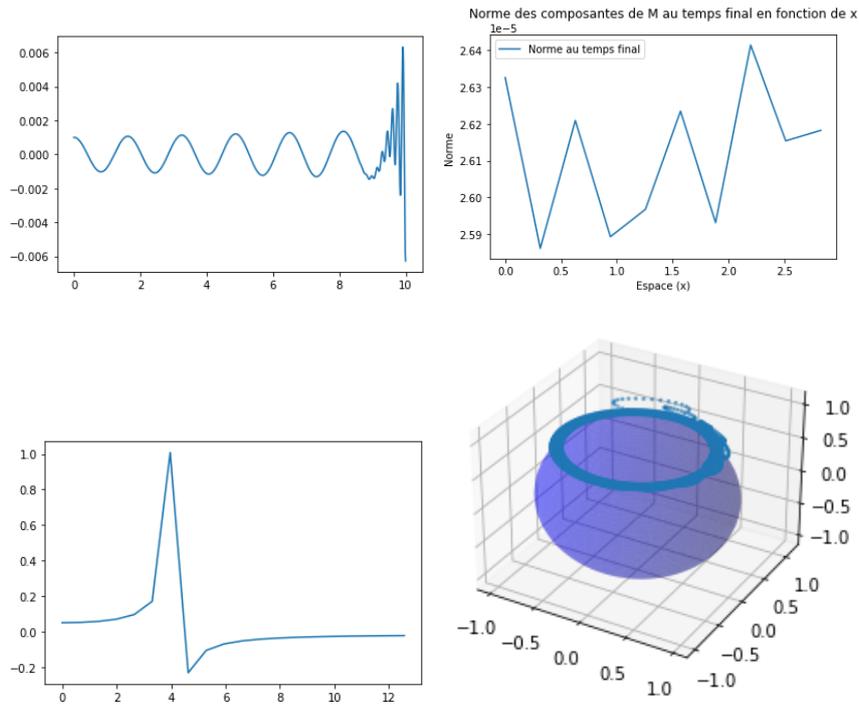


FIGURE 3 – Composante 1 en fonction du temps et Norme de $M - 1$ au temps N_t Euler explicite $k = 2$, $\epsilon = 0.001$, $a = 1$, $b = 0$, $N_x = 10$, $N_t = 2000$

de 10^{-5} . On remarque de plus que la norme sort de la sphère avec le temps ce qui nous confirme les propriétés de non-conservation géométrique.

Euler exponentiel

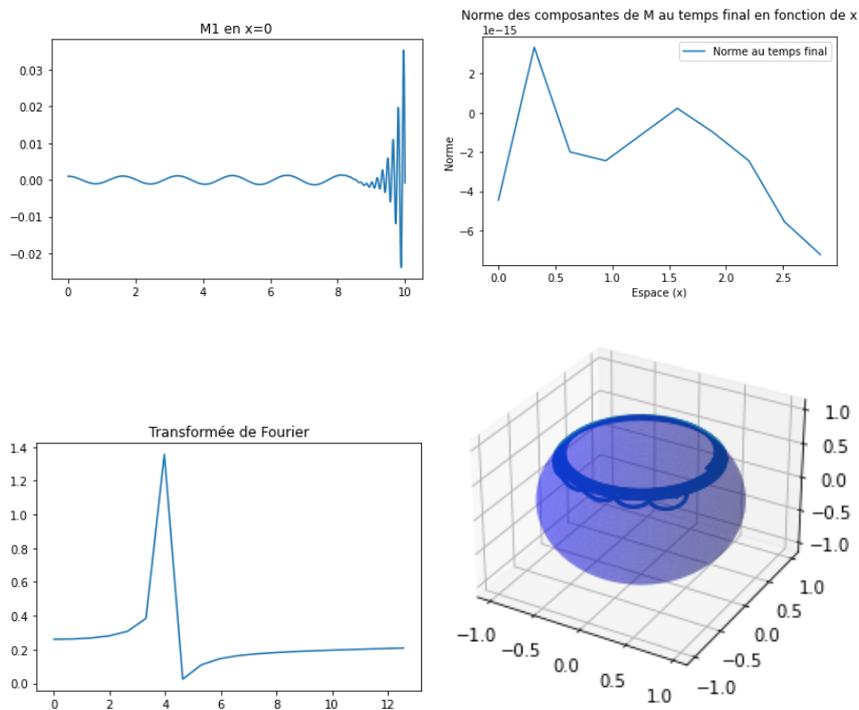


FIGURE 4 – Composante 1 en fonction du temps et Norme de $M - 1$ au temps N_t Euler exponentiel $k = 2$, $\epsilon = 0.001$, $a = 1$, $b = 0$, $N_x = 10$, $N_t = 2000$

Comme pour Euler explicite on remarque une explosion au temps final qui nous indique un

pas de temps trop élevé et une transformée de Fourier cohérente. Cependant, on a une meilleure préservation de la norme puis qu'elle est de l'ordre de 10^{-15} .

On prefera donc utiliser la méthode d'Euler exponentiel plutôt que les méthodes implicite et explicite pour le reste de l'étude.

Splitting 1

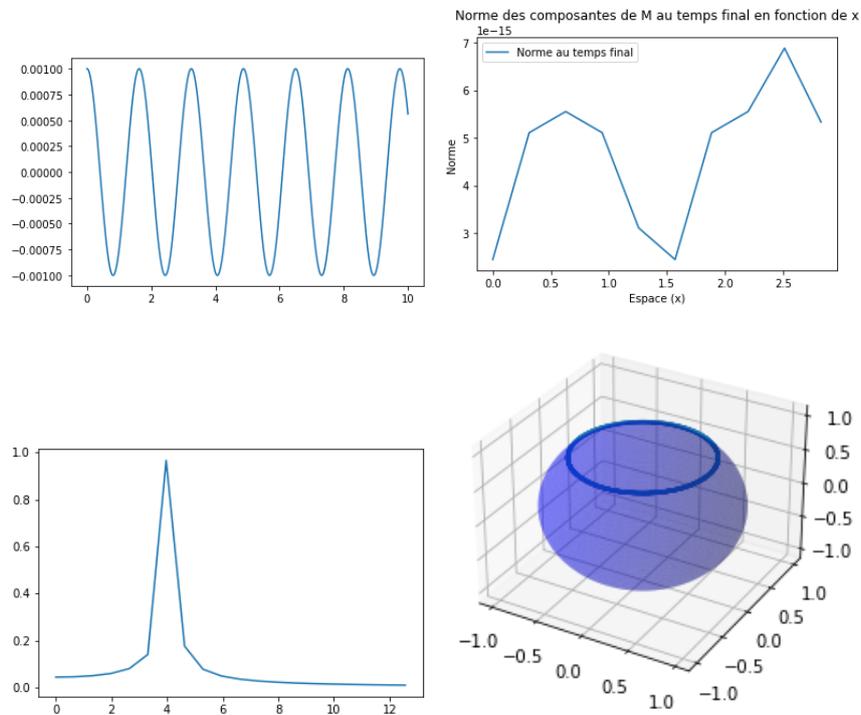
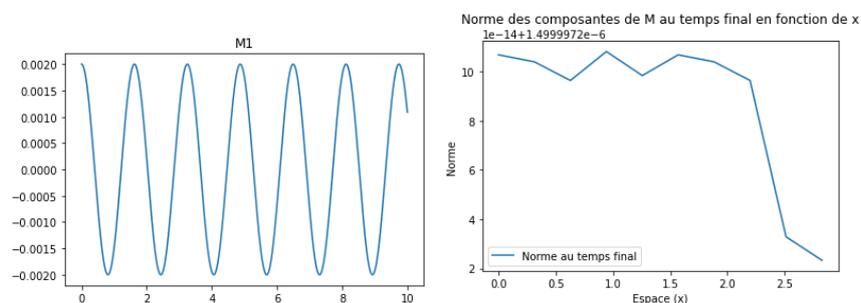


FIGURE 5 – Composante 1 en fonction du temps et Norme de $M - 1$ au temps N_t Splitting 1 $k = 2$, $\epsilon = 0.001$, $a = 1$, $b = 0$, $N_x = 10$, $N_t = 2000$

Avec notre premier splitting on a une bonne approximation de la solution avec un cosinus régulier pour la première composante, la norme de m qui est bien préservée, un pic parfaitement situé sur la fréquence ω correspondant à nos paramètre d'étude et finalement une bonne représentation sur la sphère.

Splitting 2



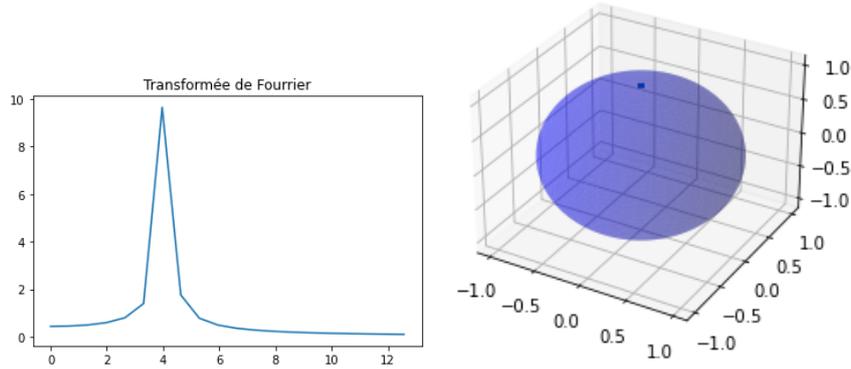


FIGURE 7 – Composante 1 en fonction du temps et Norme de $M - 1$ au temps N_t Splitting 1 $k = 2$, $\epsilon = 0.001$, $a = 1$, $b = 0$, $N_x = 10$, $N_t = 2000$

Le deuxième splitting nous a laissé perplexe car de prime abord il semble fonctionner aussi bien que le premier splitting car la transformée de Fourier correspond, la norme est bien conservée et la première composante semble se comporter comme un cosinus. Cependant, sur la sphère on ne remarque pas de mouvement. On a donc supposé que ça tournait à une très petite échelle. Après avoir essayé de debugger nous nous sommes rendus compte que dans le splitting vert, le terme $m_1 \partial_x^2 m_2 - m_2 \partial_x^2 m_1$ s'annulait. En effet, si on reprend notre condition initiale $m(0, x) = \begin{pmatrix} \frac{\epsilon \cos(kx)}{\sqrt{1+\epsilon^2}} \\ \frac{\epsilon \sin(kx)}{\sqrt{1+\epsilon^2}} \\ \frac{1}{\sqrt{1+\epsilon^2}} \end{pmatrix}$ on remarque bien qu'en réalité $m_1 \partial_x^2 m_2 = \partial_x^2 m_1$ à cause du cosinus et du sinus. Ceci rend alors la valeur de b inutile dans le splitting vert.

Néanmoins sa valeur devrait impacter le splitting rouge et bleu mais lorsque nous avons changé la valeur de b , nous avons vu que celle-ci n'impactait pas le résultat ou alors l'impactait à une échelle moindre de l'ordre de 10^{-7} . Nous n'avons pas su trouver d'explication à ce problème avant la fin de mon stage même si mon tuteur a lui aussi codé le deuxième splitting pour s'assurer que l'erreur ne provenait pas de mon programme. Nous avons aussi essayé de changer de condition initiale pour voir s'il n'y avait pas de perturbation à l'ordre supérieur mais nous n'avons rien vu.

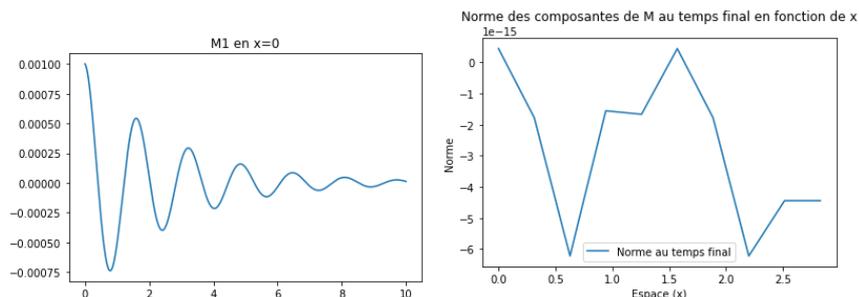
Conclusion

Après cette première étude sans amortissement, on choisira d'exclure les méthodes d'Euler implicite et explicite à cause de leur non-conservation de la norme. On gardera la méthode d'Euler exponentiel même si celle-ci nécessite un pas de temps beaucoup plus petit que le premier splitting pour bien approximer la solution. Néanmoins elle permet la conservation des propriétés géométriques, c'est pourquoi nous choisissons de la garder.

Finalement, nous avons exclu l'utilisation du splitting 2 en raison des erreurs qu'il génère.

2.4.2 Avec amortissement

Euler exponentiel



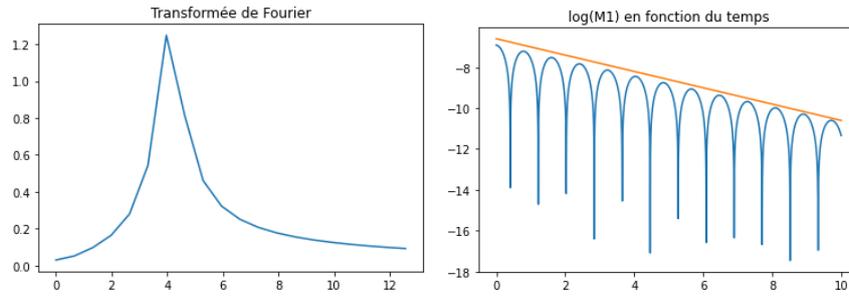


FIGURE 9 – Euler exponentiel $K = 2$, $T = 10$, $a = -1$, $b = -0.1$ $\epsilon = 0.001$ $N_t = 10000$ et $N_x = 10$

En prenant un pas de temps très petit ($\Delta t = 0,001$), on obtient une bonne approximation de la première composante sur laquelle on constate visuellement l'effet de l'amortissement. La norme est bien préservée et le pic de la transformée de Fourier se situe à la valeur correspondante de k . On peut aussi tracer le log de la première composante en $x = 0$ pour vérifier le coefficient de la pente qui est supposé correspondre à la partie imaginaire de ω .

$\omega = ak^2 - ibk^2$ on calcul manuellement ω et on obtient :

$$\omega = (-1)2^2 - i(-0,1)2^2$$

$$\omega = -4 + 0,4i$$

On a donc une pente de coefficient 0,4 ce qui semble cohérent avec la droite affichée sur le graphique.

Splitting 1

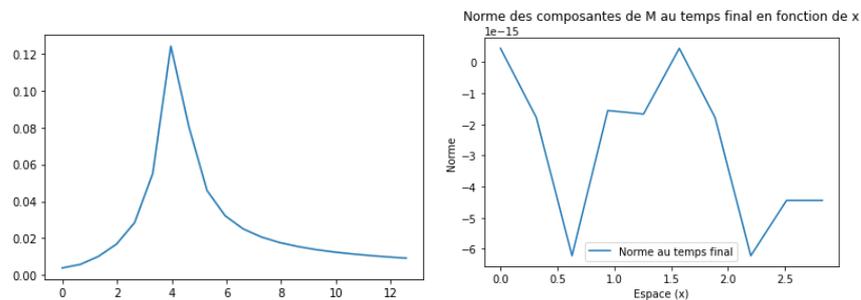
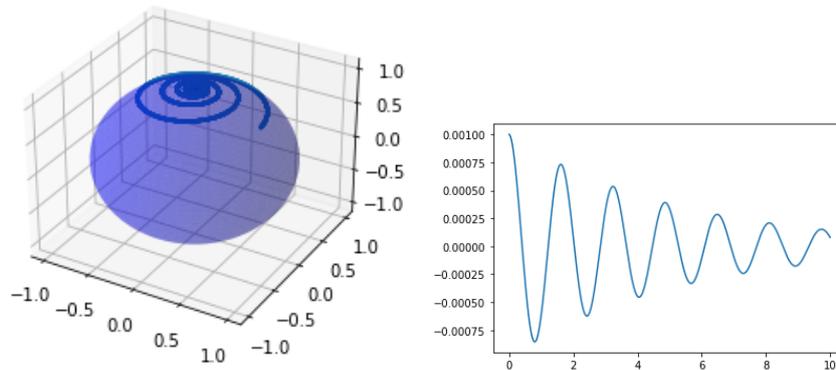


FIGURE 10 – Transformée de Fourier et Norme Splitting 1 $K = 2$, $T = 10$, $a = -1$, $b = -0.1$ $\epsilon = 0.001$ $N_t = 10000$ et $N_x = 10$



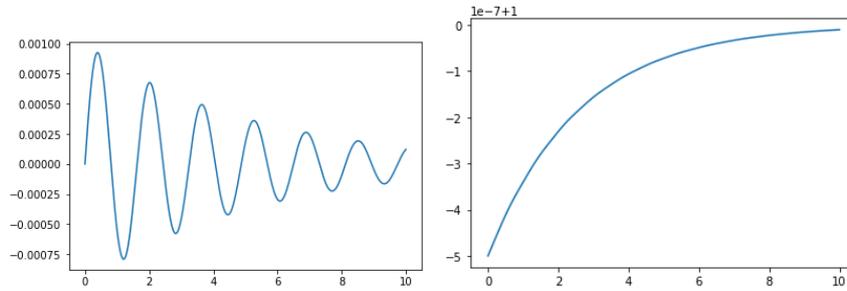


FIGURE 12 – Splitting $K = 2$, $T = 10$, $a = -1$, $b = -0.1$ $\epsilon = 0.001$ $N_t = 10000$ et $N_x = 10$

Ci-dessus, les 3 composantes principales avec le mouvement qu’elles génèrent sur la sphère. Ainsi, avec les même paramètre utilisé précédemment pour euler exponentiel, on obtient un résultat tout aussi convaincant.

Euler exponentiel et premier splitting

Afin de comparer les différentes méthodes nous avons affiché la première composante de M en $x=0$ en fonction du temps.

Ici $K = 2$, $T = 10$, $a = -1$, $b = -0.1$ et $\epsilon = 0.001$

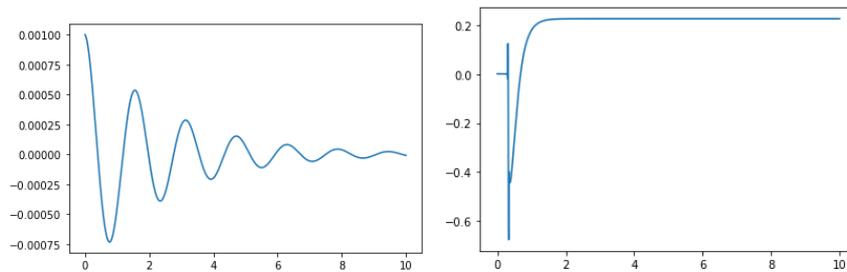


FIGURE 13 – Splitting 1 et Euler exponentiel pour $N_t = 2000$ et $N_x = 30$

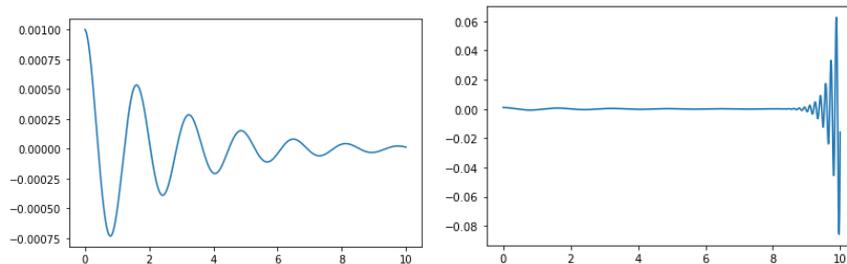


FIGURE 14 – Splitting 1 et Euler exponentiel pour $N_t = 1000$ et $N_x = 10$

Comme évoqué plus tôt dans le rapport, on remarque que pour un pas de temps pas assez petit, la méthode d’Euler exponentiel ne permet pas une approximation de la solution valide contrairement à notre première forme de splitting. On n’arrive cependant pas encore bien à expliquer pourquoi notre méthode exponentiel nécessite un pas de temps aussi petit. On suspecte une condition CFL.

Néanmoins, la manière dont nous avons codé le premier splitting semble être assez couteuse mais cela génère des resultats valide peu importe le pas de temps que ce soit avec ou sans amortissement.

2.5 Conclusion

Lors de ce stage, j'aurais pu m'initier au domaine des spin notamment grâce à certaines réunion avec des membres de l'équipe de Strasbourg afin de mieux appréhender les phénomènes physiques pour les approcher au mieux par les mathématiques.

J'ai pu tester différentes méthodes numériques et constater leurs avantages et inconvénients. A la fin de ce stage, nous avons conclut que la méthode d'Euler exponentielle et de notre premier splitting étaient les plus adaptées à ce phénomène même si elle présente chacune leurs défauts. Le splitting semble néanmoins plus fiable même si de la manière dont je l'ai programmé il est assez couteux. La méthode d'Euler exponentielle est elle aussi fiable à condition de saisir un très petit pas de temps.

Finalement, une suite possible aurait été de comprendre le problème du deuxième splitting et de le résoudre mais aussi de comprendre pourquoi la méthode d'Euler exponentielle nécessite un si petit pas de temps.

Grille de déroulement du stage

- **Semaine 1 du 02/01/24 au 08/01/2024** : Appropriation des documents déjà existants et compréhension du sujet. Rédaction du rapport.
- **Semaine 2 du 02/01/24 au 15/01/2024** : Appropriation des documents déjà existants et compréhension du sujet. Rédaction du rapport.
- **Semaine 3 du 15/01/24 au 22/01/2024** : Recherche d'une formulation mathématique permettant l'étude numérique. Rédaction du rapport.
- **Semaine 4 du 22/01/24 au 29/01/2024** : Programmation des méthodes numériques d'Euler pour l'équation sans amortissement. Rédaction du rapport.
- **Semaine 5 du 29/01/24 au 05/02/2024** : Découverte et implémentation du premier splitting. Rédaction du rapport.
- **Semaine 6 du 05/02/24 au 12/02/2024** : Recherche d'une formulation mathématique en ajoutant le terme d'amortissement. Rédaction du rapport.
- **Semaine 7 du 12/02/24 au 16/02/2024** : Recherche d'une formulation mathématique en ajoutant le terme d'amortissement et implémentation. Rédaction du rapport.
- **Semaine 8 du 19/08/24 au 26/08/2024** : Implémentation du deuxième splitting et recherche du problème de la méthode. Rédaction du rapport.
- **Semaine 9 du 26/08/24 au 02/09/2024** : Recherche du problème du deuxième splitting. Rédaction du rapport.
- **Semaine 10 du 02/08/24 au 06/09/2024** : Rédaction du rapport.

Références

- [Bañ04] L'ubomír BAÑAS. “Numerical methods for the Landau-Lifshitz-Gilbert equation”. In : *International Conference on Numerical Analysis and Its Applications*. Springer. 2004, p. 158-165.
- [Hai+06] Ernst HAIRER et al. “Geometric numerical integration”. In : *Oberwolfach Reports 3.1* (2006), p. 805-882.
- [Rug16] Michele RUGGERI. “Coupling and numerical integration of the Landau-Lifshitz-Gilbert equation”. Thèse de doct. Technische Universität Wien, 2016.
- [Son16] Eric SONNENDRÜCKER. “Numerical methods for the Vlasov-Maxwell equations”. In : *Book manuscript to be published with Springer* (2016).
- [BCM24] Sergio BLANES, Fernando CASAS et Ander MURUA. “Splitting methods for differential equations”. In : *arXiv preprint arXiv :2401.01722* (2024).