

Master Calcul Scientifique et Modélisation
2025-2026

Université de Rennes

Intégration numérique de dynamiques sur groupes de Lie matriciels

Marie-Clémence Guéry

Encadrants : Adrien Busnot Laurent & François Castella



Année universitaire 2025–2026

Remerciements

Je tiens tout d'abord à remercier chaleureusement l'Université de Rennes et l'équipe pédagogique de la formation pour leur accompagnement et leur soutien tout au long de l'année. Leurs enseignements et leurs conseils m'ont permis d'aborder sereinement ce Master 1 et de décrocher rapidement ce stage.

Mes plus sincères remerciements s'adressent à mes encadrants de stage, Adrien Busnot Laurent et François Castella. Je les remercie profondément pour le temps qu'ils m'ont accordé, leur présence constante, ainsi que leur bienveillance tout au long de ce projet. Leurs conseils avisés et leur rigueur scientifique m'ont grandement guidée dans la réalisation de mes travaux.

Résumé

Les méthodes d'intégration numérique classiques (Euler, Runge-Kutta), qui sont vectorielles, ne préservent pas la structure géométrique des équations différentielles posées sur des groupes de Lie matriciels : nous le vérifions expérimentalement sur $SO(3)$, où la solution quitte le groupe dès les premières itérations. Ce travail étudie et implémente sous MATLAB plusieurs méthodes intrinsèques : Lie-Euler, Magnus, Runge-Kutta Munthe-Kaas (RKMK) et méthodes sans commutateurs (Crouch-Grossman, CF3, CF4) construites à partir de l'application exponentielle pour garantir que la solution numérique reste sur le groupe, avec pour application la simulation de trajectoires sur la sphère $S^2 := SO(3)/SO(2)$.

Les ordres de convergence théoriques sont validés numériquement par régression log-log de l'erreur pour chaque méthode. Nous montrons en particulier que la troncature de la série de Baker à l'ordre 2 limite la méthode Magnus-Simpson à l'ordre 2 malgré une quadrature d'ordre 4, et que la méthode CF4 atteint l'ordre 4 en ne requérant que 6 exponentielles de matrices contre 15 pour RKMK4, pour un gain de temps de calcul d'un facteur 4 environ. Ces résultats confirment l'intérêt des méthodes sans commutateurs pour la simulation géométrique sur $SO(3)$, avec des applications potentielles dans les corps célestes.

Mots-clés : intégration géométrique, groupes de Lie, $SO(3)$, série de Baker, méthodes de Runge-Kutta Munthe-Kaas, méthodes sans commutateurs.

Table des matières

1	Introduction	4
1.1	Groupe de Lie et algèbre de Lie	4
1.2	La sphère S^2 : une variété homogène	5
1.3	Motivations numériques	6
2	EDO à coefficients constants sur groupe de Lie	7
3	EDO à coefficients dépendant du temps sur groupe de Lie	10
3.1	Théorie de Magnus	10
3.2	Illustrations numériques	11
4	EDO à coefficients dépendant de la position sur groupe de Lie	16
4.1	Méthodes de Runge-Kutta Munthe-Kaas :	16
4.2	Méthodes sans commutateurs :	22
5	Conclusion	30

1 Introduction

La simulation numérique de systèmes dynamiques joue un rôle crucial dans de nombreux domaines de la science et de l'ingénierie, allant de la mécanique céleste à la robotique. Traditionnellement, les équations différentielles ordinaires (EDO) sont discrétisées à l'aide de méthodes numériques classiques telles que les méthodes d'Euler ou Runge-Kutta.

Cependant, lorsque l'espace étudié n'a plus de structure géométrique linéaire (comme une variété différentielle ou un groupe de Lie), ces méthodes échouent fréquemment à préserver les invariants structurels du système.

Ce travail s'inscrit dans le cadre de l'intégration numérique géométrique, une partie de l'analyse numérique qui conçoit des algorithmes capables de préserver les propriétés géométriques des équations.

Dans un premier temps, nous mettrons en évidence l'échec des méthodes standards (Euler Explicite, RK4) qui s'écartent instantanément de la variété $SO(3)$. Ensuite, nous introduirons les méthodes de Lie-Euler et Magnus. Ces approches reposent sur l'application exponentielle pour garantir la structure géométrique. Nous comparerons l'ordre de convergence de différentes techniques d'approximation (Rectangle, Point Milieu, Simpson). Suite à cela, nous aborderons le cas le plus général où le champ de vecteurs dépend de la position courante. Nous étudierons et implémenterons les méthodes de Runge-Kutta Munthe-Kaas (RKMK), avant d'explorer les méthodes sans commutateurs telles que les schémas Crouch-Grossman et les méthodes CF3/CF4, conçues pour réduire le coût de calcul.

1.1 Groupe de Lie et algèbre de Lie

Définition 1.1 (Variété différentielle [14]). *Soit M un espace topologique.*

- *Une carte de dimension n sur M est un homéomorphisme [16] $\phi : U \rightarrow V$ où U est un ouvert de M et V un ouvert de $\mathbb{R}^n$ [14].*
- *Deux cartes (U_1, ϕ_1) et (U_2, ϕ_2) sont dites compatibles si $U_1 \cap U_2 = \emptyset$ ou si $x \rightarrow \phi_2 \circ \phi_1^{-1}(x)$ est un difféomorphisme [17] de $\phi_1(U_1 \cap U_2)$ vers $\phi_2(U_1 \cap U_2)$ [14].*
- *Un atlas de M est une famille compatible deux à deux dont les domaines de définition recouvrent M [14].*

Une variété différentielle est un espace topologique M muni d'un tel atlas.

Définition 1.2 (groupe de Lie [1]). *Un groupe de Lie (réel) G est un groupe muni d'une structure de variété différentielle tel que la multiplication $G \times G \rightarrow G$ et l'inversion $G \rightarrow G$ soient des applications de classe $\mathcal{C}^\infty$.*

Définition 1.3 (algèbre de Lie [20]). *A chaque groupe de Lie est associée une algèbre de Lie qui est un espace vectoriel, notée $\mathfrak{g}$ de dimension finie qui est munie d'une*

application bilinéaire antisymétrique

$$[\cdot, \cdot] : \mathfrak{g} \times \mathfrak{g} \rightarrow \mathfrak{g}$$

vérifiant l'identité de Jacobi :

$$\forall x, y, z \in \mathfrak{g}, [x, [y, z]] + [y, [z, x]] + [z, [x, y]] = 0.$$

avec $[A, B] = AB - BA$ pour l'algèbre des matrices.

On peut rassembler certains groupes de Lie avec leur algèbre associée :

Groupe de Lie	Algèbre de Lie
$GL(n) = \{Y \in \mathbb{R}^{n \times n} \mid \det Y \neq 0\}$	$\mathfrak{gl}(n) = \mathbb{R}^{n \times n}$
$SL(n) = \{Y \in \mathbb{R}^{n \times n} \mid \det Y = 1\}$	$\mathfrak{sl}(n) = \{Y \in \mathbb{R}^{n \times n} \mid \text{tr}(Y) = 0\}$
$O(n) = \{Y \in \mathbb{R}^{n \times n} \mid Y^T Y = I\}$	$\mathfrak{o}(n) = \{Y \in \mathbb{R}^{n \times n} \mid Y^T + Y = 0\}$
$SO(n) = \{Y \in O(n) \mid \det Y = 1\}$	$\mathfrak{so}(n) = \{Y \in \mathbb{R}^{n \times n} \mid Y^T + Y = 0\}$
$Sp(n) = \{Y \in \mathbb{R}^{n \times n} \mid Y^T J Y = J\}$	$\mathfrak{sp}(n) = \{Y \mid JY + Y^T J = 0\}$

TABLE 1 – Groupes de Lie et Algèbres de Lie associées

Définition 1.4 (exponentielle de matrice [10]). Soit $A \in \mathbb{R}^{n \times n}$. Pour toute norme matricielle, la série de matrices

$$\sum_{k=0}^{\infty} \frac{A^k}{k!}$$

est absolument convergente. On définit ainsi l'application exponentielle de matrices :

$$\mathfrak{g} \rightarrow G : A \mapsto \exp(A) = \sum_{k=0}^{\infty} \frac{A^k}{k!}.$$

Exemple 1.5. Dans le cas du groupe $SO(3)$, l'application exponentielle envoie une matrice antisymétrique de $\mathfrak{so}(3)$ sur une rotation de $SO(3)$.

1.2 La sphère S^2 : une variété homogène

Dans le cadre de ce travail, nous nous intéressons en particulier à la sphère S^2 . La sphère S^2 peut être perçue comme l'espace quotient des rotations de l'espace par le groupe des rotations dans le plan, soit :

$$S^2 := SO(3)/SO(2)$$

Définition 1.6. Soit G un groupe opérant sur un ensemble E , et x un élément de E . On appelle :

- Une action est transitive si $\forall x, y \in E, \exists g \in G, g \cdot x = y$ [30].
- L'orbite de x est la partie de E définie par $\text{Orb}(x) = \{g \cdot x; g \in G\}$ [30].
- Le stabilisateur de x est la partie de G définie par $G_x = \{g \in G; g \cdot x = x\}$ [30].

On étudie l'action géométrique du groupe des rotations $SO(3)$ sur la sphère S^2 . On fixe un point de référence (par exemple le pôle Nord $N = (0,0,1)$).

- L'action est *transitive* : En appliquant toutes les rotations de $SO(3)$ au pôle Nord N , on peut atteindre n'importe quel point de la sphère. L'ensemble de la sphère S^2 est donc décrit comme l'orbite du point N sous l'action de $SO(3)$.
- Le *stabilisateur* du pôle Nord est $SO(2)$ puisque les seules rotations de $SO(3)$ qui laissent le pôle Nord N invariant sont les rotations s'effectuant autour de l'axe vertical Z sont les rotations de $SO(2)$.

Pour autant, la sphère S^2 n'est pas un groupe de Lie. Bien qu'elle soit une variété différentielle, on ne peut pas la munir d'une structure de groupe [31].

Dans notre cas, nous allons nous intéresser à $G : SO(3)$, puisque c'est ce groupe qui nous permet de décrire les déplacements sur la sphère S^2 . Notre algèbre de Lie $\mathfrak{g}$ est donc l'espace tangent en l'élément neutre $e = I$, soit $\mathfrak{g} = T_I G = \mathfrak{so}(3)$, avec I la matrice identité.

1.3 Motivations numériques

Le but principal de ce projet est le développement de méthodes numériques permettant de simuler un déplacement directement sur la sphère S^2 . On s'inspirera notamment des approches décrites dans [13]. Les déplacements sur la sphère peuvent être décrits par des rotations dépendant de la position courante. Ces rotations sont représentées par des éléments de $SO(3)$.

Dans la suite de ce projet, nous étudierons des équations différentielles de la forme

$$y'(t) = \underbrace{A(t, y(t))}_{\in \mathfrak{so}(3)} y(t)$$

avec $y_0 \in SO(3)$.

Théorème 1.7. *Soit l'EDO $y'(t) = A(t, y(t))y(t)$ avec $A : \mathbb{R} \times SO(3) \rightarrow \mathfrak{so}(3)$ et $y_0 \in SO(3)$. A est supposée lipschitzienne en y . Alors y est définie sur $\mathbb{R}$ et $y(t) \in SO(3), \forall t \in \mathbb{R}$ [11].*

Démonstration. Soit $A(t, y)$ une matrice antisymétrique et soit $y_0 \in SO(3)$.

Posons $M(t) = y(t)y(t)^T - I_3$.

Montrons que $M(t) = 0$.

On a $\frac{d}{dt}[M(t)] = y'(t)y(t)^T + y(t)(y'(t))^T$.

Or on sait que $y'(t) = A(t, y(t))y(t)$.

Donc, $\frac{d}{dt}[M(t)] = (A(t, y(t))y(t))y(t)^T + y(t)(A(t, y(t))y(t))^T$

De ce fait, $\frac{d}{dt}[M(t)] = (A(t, y(t))y(t))y(t)^T + y(t)(y(t)^T A(t, y(t))^T)$

Comme $A(t, y(t))^T = -A(t, y(t))$:

$$\frac{d}{dt}[M(t)] = (A(t, y(t))y(t))y(t)^T - y(t)(y(t)^T A(t, y(t)))$$

$$\text{Donc } \frac{d}{dt}[M(t)] = A(t, y(t)) \cdot (M(t) + I_3) - (M(t) + I_3) \cdot A(t, y(t))$$

$$\text{Donc } \frac{d}{dt}[M(t)] = (A(t, y(t)) \cdot M(t) + A(t, y(t)) \cdot I_3) - (M(t) \cdot A(t, y(t)) + I_3 \cdot A(t, y(t)))$$

Comme l'identité I_3 commute avec toutes les matrices de taille 3, on a $\frac{d}{dt}[M(t)] = A(t, y(t)) \cdot M(t) - M(t) \cdot A(t, y(t))$.

Or la matrice nulle est une solution évidente de cette équation.

Par le théorème de Cauchy-Lipschitz (car A est de classe suffisante et localement lipschitzienne), et comme la condition initiale donne $M(0) = y_0 y_0^T - I_3 = 0$, c'est l'unique solution, donc $y(t)$ reste orthogonale.

Montrons également que $\det(y(t)) = 1$.

$$\text{On a } \det(y(t)y(t)^T) = \det(I_3) = 1$$

$$\text{Donc } (\det(y(t)))^2 = 1 \implies \det(y(t)) = \pm 1.$$

Or comme $y_0 \in SO(3)$, le déterminant de $y(t)$ vaut 1. □

2 EDO à coefficients constants sur groupe de Lie

Dans cette section, toutes les implémentations numériques ont été effectuées sur MATLAB. L'exponentielle de matrice a été calculée à l'aide de la commande `expm` qui utilise l'approximation de Padé [29], l'approximation des séries de Taylor, les valeurs propres et vecteurs propres de la matrice.

Les paramètres du problème sont définis de telle sorte :

- $t_0 = 0$
- $t_f = 2$
- $n = 500$
- $h = \frac{|t_f - t_0|}{n}$

Pour chacune des méthodes, il est nécessaire de vérifier que la matrice appartient bien à $SO(3)$. Il faut donc qu'elle appartienne au groupe orthogonal donc doit vérifier la condition $\|y \cdot y^T - I\|_{fro} < \text{tol}$ et également il est nécessaire que $|\det(y) - 1| < \text{tol}$. On utilise la norme de Frobenius ($\|A\|_F^2 = \sum_{ij} |a_{i,j}|^2$) car elle semble la plus usuelle et étant donné que nous sommes en dimension finie, toutes les normes sont équivalentes [15].

Théorème 2.1. *Si A est une constante, nous savons que la solution est de la forme*

$$y(t) = \underbrace{\exp(\underbrace{tA}_{\in \mathfrak{so}(3)})}_{\in SO(3)} \underbrace{y_0}_{\in SO(3)}.$$

Exemple 2.2. $A = \begin{pmatrix} 0 & 1 & 2 \\ -1 & 0 & 3 \\ -2 & -3 & 0 \end{pmatrix} \in \mathfrak{so}(3)$

avec $y_0 = I_3$

Dans ce cas précis, il est naturel d'utiliser des méthodes numériques standards comme les méthodes de Runge-Kutta [12] :

— Euler Explicite : $y_{n+1} = (I + hA)y_n$

Euler Explicite : Calculer $y_{n+1} = (I + hA)y_n$

Require: $n > 0$

Ensure: $y_{n+1} = (I + hA)y_n$

$h = \frac{|t_f - t_0|}{N}$

$y \leftarrow y_0$

for $k = 1 \rightarrow N$ **do**

if $|y^T y - I| > \epsilon$ ou $|\det(y) - 1| > \epsilon$ **then**

y n'appartient plus à $SO(3)$

 break

else

$y \leftarrow (I + hA)y$

end if

end for

— Runge-Kutta 4 : $y_{n+1} = P(hA)y_n$

Runge-Kutta 4 : Calculer $y_{n+1} = P(hA)y_n$

Require: $n > 0$

Ensure: $y_{n+1} = P(hA)y_n$

$h = \frac{|t_f - t_0|}{N}$

$y \leftarrow y_0$

for $k = 1 \rightarrow N$ **do**

if $|y^T y - I| > \epsilon$ ou $|\det(y) - 1| > \epsilon$ **then**

y n'appartient plus à $SO(3)$

 break

else

$k_1 \leftarrow Ay$

$k_2 \leftarrow A(y + \frac{h}{2}k_1)$

$k_3 \leftarrow A(y + \frac{h}{2}k_2)$

$k_4 \leftarrow A(y + k_3h)$

$y \leftarrow y + h(\frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6})$

end if

end for

Après implémentation, on remarque qu'après un certain nombre d'itérations, les matrices n'appartiennent plus à $SO(3)$.

Effectivement, pour $n = 500$, et $t_f = 2$ lorsque nous représentons le logarithme décimal de la norme de Frobenius de la différence entre $y \cdot y^T$ et I , nous obtenons la figure 1 suivante :

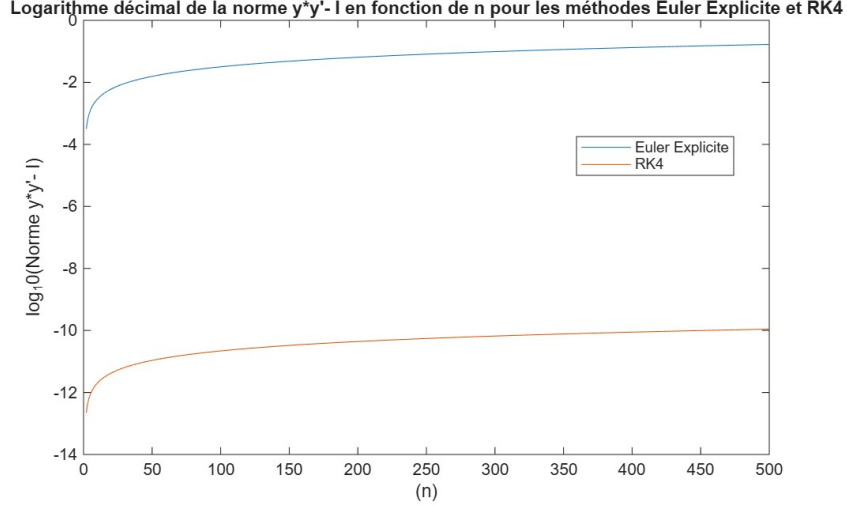


FIGURE 1 – Graphe du logarithme de la norme de $yy^T - I$ en fonction de n pour les méthodes Euler Explicite et RK4

Notre tolérance étant initialisée à 10^{-12} , on remarque bien que sur la figure 1, dès la première itération, la norme de $yy^T - I$ pour la méthode de Euler Explicite est supérieure à la tolérance, donc la solution n'est plus sur la sphère. Quant à la méthode de RK4, c'est au bout de 5 itérations que la solution n'appartient plus à $SO(3)$.

Nous pouvons par exemple représenter la solution y calculée par la méthode d'Euler Explicite qui permet de voir que la solution sort de la sphère. La solution y étant une matrice, on affiche les 3 colonnes sur la sphère (rouge : première colonne, vert : deuxième colonne et bleu : troisième colonne).

Pour un temps final égal à 20 et une subdivision de 5000 intervalles, nous obtenons ce graphe :

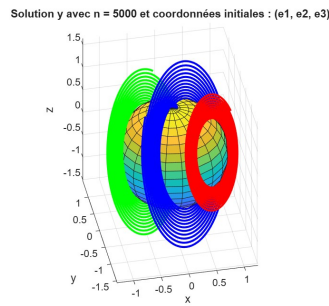


FIGURE 2 – Graphe de la solution y avec $t_f = 20$ et $n = 5000$ pour la méthode d'Euler Explicite

Ces deux méthodes ne sont donc pas performantes pour notre cas, puisqu'elles ne préservent pas la structure géométrique. En effet, ce sont des méthodes vectorielles [18], et comme la sphère n'est pas un espace vectoriel (car une combinaison linéaire de deux éléments n'appartient généralement plus à la sphère) [32], les deux méthodes ne permettent pas de rester sur la sphère.

On pourrait penser que faire une projection sur la sphère permet de résoudre le problème. Mais le fait de projeter revient à effectuer une approximation à l'ordre 1 de l'application exponentielle (une rétraction)[33], ce qui dégrade l'ordre de convergence des schémas d'ordre supérieur. Si on utilise une méthode d'ordre 4 (comme RK4) et qu'à chaque étape on projette brutalement sur la sphère via une méthode d'ordre 1, l'erreur globale finale va être dominée par l'erreur de projection (ordre 1). On perd tout l'intérêt mathématique de l'ordre 4. Cela justifie l'intérêt des méthodes intrinsèques (Magnus, RKMk, Crouch-Grossman, CF) qui forment le schéma directement dans l'algèbre de Lie via l'application exponentielle, s'affranchissant ainsi de toute étape de projection.

3 EDO à coefficients dépendant du temps sur groupe de Lie

Lorsque la matrice A dépend du temps t , la résolution de l'équation différentielle linéaire $y'(t) = A(t)y(t)$ devient plus complexe que dans le cas autonome. En effet, dès lors que les matrices $A(t)$ prises à différents instants ne commutent pas, la simple exponentielle de l'intégrale de $A(t)$ ne fournit plus la solution exacte du système.

Pour surmonter cette difficulté tout en garantissant que la solution approchée reste confinée dans le groupe de Lie, Magnus a introduit une nouvelle approche [19]. L'idée directrice consiste à exprimer la solution sous la forme d'une unique exponentielle de matrice, $y(t) = \exp(\Omega(t))y_0$, où l'opérateur $\Omega(t)$ est défini par une série infinie d'intégrales de commutateurs imbriqués.

Toutefois, la manipulation analytique de cette série devient rapidement compliquée. C'est pourquoi de nombreux travaux numériques se sont attachés à concevoir des approximations efficaces et à optimiser le calcul de ces termes, notamment à travers les algorithmes développés par Blanes [3]. Aujourd'hui, ces méthodes de Magnus se classent parmi les outils les plus puissants de l'intégration numérique géométrique sur les groupes de Lie, comme le souligne l'étude approfondie d'Iserles. [13]. La section suivante détaille les fondements théoriques de cette approche.

3.1 Théorie de Magnus

Si A ne dépend que de t , on a

$$y'(t) = A(t)y(t).$$

Théorème 3.1. *Si $[A(t), A(s)] = 0 \ \forall t, s$, alors la solution de l'équation différentielle*

$y'(t) = A(t)y(t)$ avec la condition initiale $y(0) = y_0$ est [8] :

$$y(t) = \exp \left(\int_0^t A(s) ds \right) y_0.$$

Question : Peut-on écrire $y(t) = \exp(\Omega(t))y_0$?

Lemme 3.2. $y'(t) = (\frac{d}{dt} \exp(\Omega(t)))\dot{\Omega}(t)y_0 = (d \exp_{\Omega(t)}(\dot{\Omega}(t))) \exp(\Omega(t))y_0$ [12].

Proposition 3.3. Si $\Omega(t)$ vérifie l'équation $\Omega'(t) = (d \exp_{\Omega(t)})^{-1} A(t)$, alors la solution de $y'(t) = A(t)y(t)$ est donnée par $y(t) = \exp(\Omega(t))y_0$.

Démonstration. Par hypothèse, on a $\Omega'(t) = (d \exp_{\Omega(t)})^{-1}(A(t))$.

Donc $d \exp_{\Omega(t)}(\Omega'(t)) = A(t)$.

On définit $y(t) = \exp(\Omega(t))y_0$.

Donc $y'(t) = \frac{d}{dt}(\exp(\Omega(t)))y_0$.

Donc d'après le lemme 3.2, $y'(t) = (d \exp_{\Omega(t)}(\Omega'(t)) \exp(\Omega(t)))y_0$.

Donc $y'(t) = (A(t) \exp(\Omega(t)))y_0$.

Or $\exp(\Omega(t))y_0 = y(t)$.

Donc $y'(t) = A(t)y(t)$. □

Lemme 3.4. $(d \exp_{\Omega(t)})^{-1} = \sum_{k=0}^{\infty} \frac{B_k}{k!} ad_{\Omega(t)}^k$ [12] où :

- B_k désigne les nombres de Bernoulli ($B_0 = 1, B_1 = -\frac{1}{2}, B_2 = \frac{1}{6}, B_3 = 0, \dots$).
- $ad_{\Omega}(A) = [\Omega, A] = \Omega A - A \Omega$ représente l'opérateur adjoint.

3.2 Illustrations numériques

Dans cette section, nous allons nous intéresser à la méthode de Lie Euler, ainsi que celle de Magnus.

Définition 3.5 (Méthode Lie-Euler). $y_{n+1} = \exp(hA(t_n))y_n$

Lie Euler : Calculer $y_{n+1} = \exp(hA(t_n))y_n$

Require: $n > 0$ **Ensure:** $y_{n+1} = \exp(hA(t_n))y_n$

$$h = \frac{|t_f - t_0|}{N}$$

 $y \leftarrow y_0$ **for** $k = 1 \rightarrow N$ **do****if** $|y^T y - I| > \epsilon$ ou $|det(y) - 1| > \epsilon$ **then** y n'appartient plus à $SO(3)$ **break****else**

$$t_k \leftarrow t_0 + (k - 1)h$$

$$y \leftarrow \exp(hA(t_k))y$$

end if**end for**

Définition 3.6 (Méthode de Magnus). $y_{n+1} = \exp(h\Omega(t_n))y_n$

Comme dit précédemment, l'utilisation de l'exponentielle permet de rester sur la sphère. Il est donc intéressant d'exprimer la solution y avec une exponentielle, pour pouvoir utiliser des outils intrinsèques à notre espace géométrique.

Pour cette méthode, il faut que l'on sache déterminer Ω_2 .

D'après Baker [12], nous avons

$$\Omega'(t) = \sum_{k=0}^{\infty} \frac{B_k}{k!} ad_{\Omega(t)}^k A(t).$$

Dans ce cas, nous allons nous arrêter à l'ordre 2 puisque cela semble être le plus simple et le moins embêtant.

$$\text{Donc } \Omega'(t) \approx A(t) - \frac{1}{2}[\Omega, A(t)].$$

$$\text{D'où } \Omega(t) \approx \int_0^t A(x) dx - \int_0^t \frac{1}{2}[\Omega(x), A(x)] dx.$$

Donc on retrouve une équation de la forme :

$$\Omega(t) = F(\Omega(t)).$$

Pour cela, nous allons utiliser la méthode du point-fixe de Banach-Picard.

$$\Omega_0(t) = 0$$

$$\Omega_1(t) = \int_0^t A(x) dx$$

$$\Omega_2(t) = \int_0^t A(x) dx - \frac{1}{2} \int_0^t [\int_0^x A(\sigma) d\sigma, A(x)] dx$$

Afin de trouver Ω_2 , nous allons utiliser plusieurs méthodes numériques afin de l'approximer.

- Rectangle à gauche : $\int_0^t A(x) dx = \sum_{k=0}^{N-1} A(t_k)h$
- Point milieu : $\int_0^t A(x) dx = \sum_{k=0}^{N-1} A(t_{k+1/2})h$
- Simpson : $\int_0^t A(x) dx = \sum_{k=0}^{N-1} \frac{h}{6} [A(x_k) + 4A(x_{k+1/2}) + A(x_{k+1})]$

Magnus : Calculer $y_{n+1} = \exp(h\Omega_2(t_n))y_n$

Require: $n > 0$

Ensure: $y_{n+1} = \exp(h\Omega_2(t_n))y_n$

$$h = \frac{|t_f - t_0|}{N}$$

$y \leftarrow y_0$

for $k = 1 \rightarrow N$ **do**

if $|y^T y - I| > \epsilon$ ou $|\det(y) - 1| > \epsilon$ **then**

y n'appartient plus à $SO(3)$

break

else

$\Omega_2 \leftarrow 0$

$t_k \leftarrow t_0 + (k-1)h$

 approximation de Ω_2 via la méthode X

$y \leftarrow \exp(\Omega_2)y$

end if

end for

Exemple 3.7. $y(t) \in SO(3)$ avec $y_0 = I_3$

$$A(t) = \begin{pmatrix} 0 & \cos(2t) & \sin(t) \\ -\cos(2t) & 0 & \cos(9t) \\ -\sin(t) & -\cos(9t) & 0 \end{pmatrix} \in \mathfrak{so}(3)$$

Pour étudier les conditions d'appartenance à $SO(3)$, les paramètres du problème sont définis de telle sorte :

- $t_0 = 0$
- $t_f = 1000$
- $h = \frac{|t_f - t_0|}{n}$
- $n = 50000$

Pour étudier l'ordre de convergence, les paramètres du problème sont définis de telle sorte :

- $t_0 = 0$
- $t_f = 2$
- $h = \frac{|t_f - t_0|}{n}$
- $n \in [10, 50, 250, 500, 700, 1000]$

Pour chaque méthode, afin de comparer la solution calculée avec la solution la plus fidèle, on considère la solution finale de notre méthode avec $n = 10000$, et on utilise la norme de Frobenius ($\|A\|_F^2 = \sum_{i,j} |a_{i,j}|^2$) de la différence entre la solution finale avec $n = 10000$ et pour chaque n considéré.

Pour chacune des méthodes, il est nécessaire de vérifier que la matrice appartient bien à $SO(3)$, pour cela on utilise la condition $|det(y) - 1| < tol$ et $|y.y^T - I|_F < tol$. On utilise la norme de Frobenius car elle semble la plus usuelle et étant donné que nous sommes en dimension finie, toutes les normes sont équivalentes.

Après implémentation, nous retrouvons ces 3 graphes.

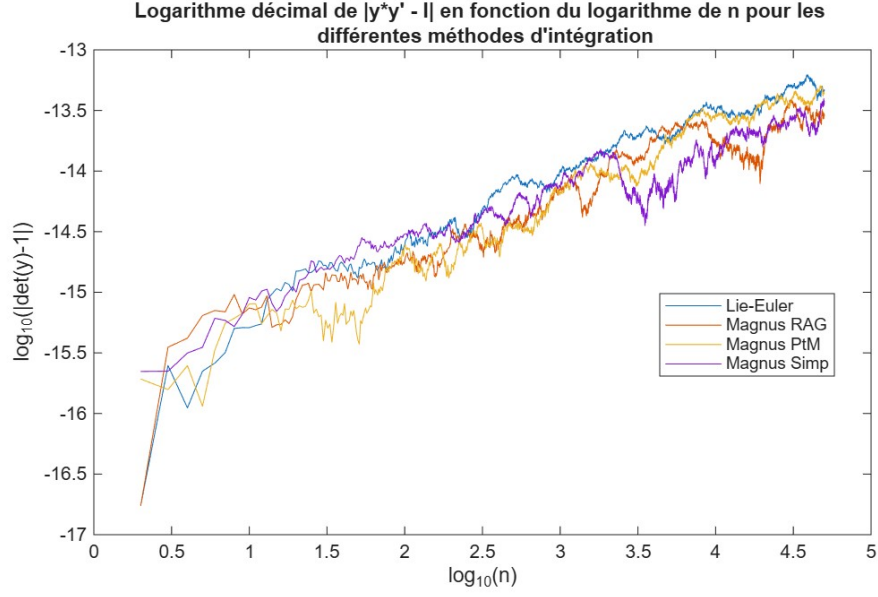


FIGURE 3 – Logarithme décimal de $|yy^t - I|$ des méthodes Lie-Euler et Magnus

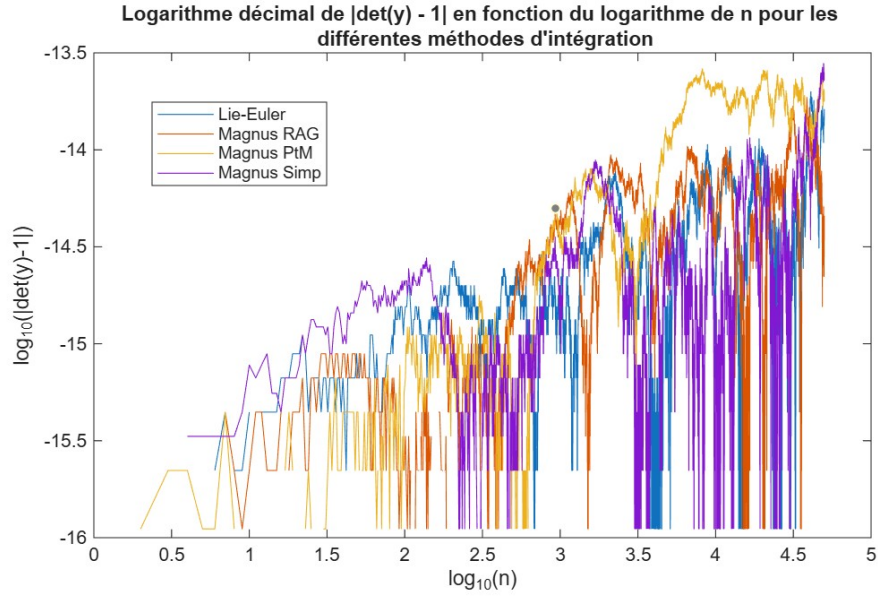


FIGURE 4 – Logarithme décimal de $|det(y) - 1|$ des méthodes Lie-Euler et Magnus

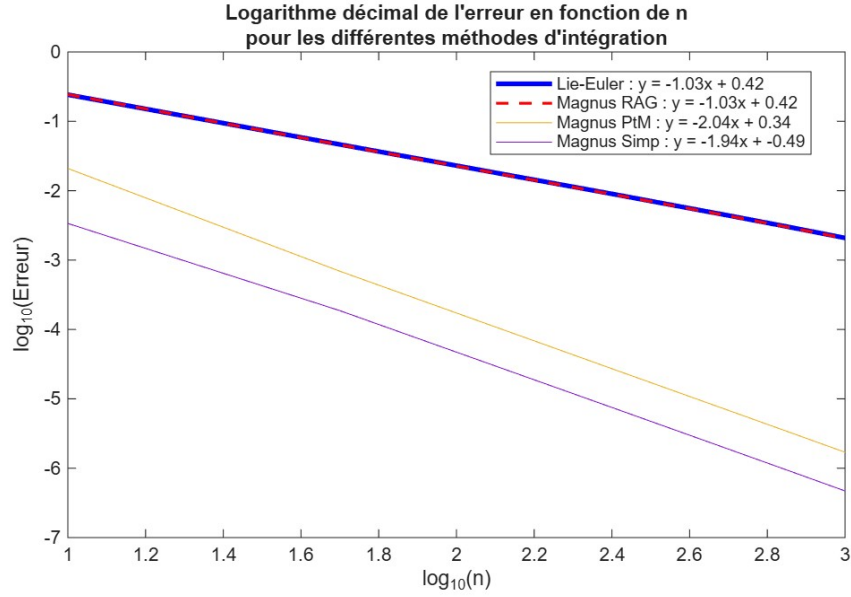


FIGURE 5 – Courbes de convergence de l'erreur des méthodes Lie-Euler et Magnus

Sur la figure 3, on remarque que le logarithme décimal de $|yy^t - I|$ de chaque méthode est inférieur à notre tolérance placée à 10^{-12} . On observe également de grosses oscillations qui sont causés par les erreurs d'arrondis dûes au zéro-machine.

Sur la figure 4, on remarque que le logarithme décimal de $|det(y) - 1|$ de chaque méthode est inférieur à notre tolérance placée à 10^{-12} .

Avec ces deux conditions respectées, nous pouvons confirmer que les méthodes Lie-Euler et Magnus permettent de rester sur la sphère et respectent donc la structure géométrique du problème.

Sur la figure 5, on retrouve les différents ordres de convergence des deux méthodes.

Effectivement, pour la méthode Lie-Euler, nous avons l'équation suivante de notre courbe : $\log_{10}(Err) = -1.03 \cdot \log_{10}(n) + 0.42$, nous avons $Err = n^{-1.03} \cdot 10^{0.42}$ donc $Err \approx \frac{1}{n} \cdot C$.

Donc l'ordre pour la méthode de Lie-Euler est 1. En suivant le même raisonnement, pour la méthode de Magnus, pour les différentes méthodes d'approximations, nous retrouvons ces différents ordres :

- Magnus Rectangle à gauche : 1
- Magnus Point Milieu : 2
- Magnus Simpson : 2

L'obtention de ces ordres est cohérente car la méthode de rectangle à gauche est d'ordre 1 et celle du Point Milieu est d'ordre 2. Quant à la méthode de Simpson, elle est d'ordre 4, mais rappelons que notre série de Baker a été tronquée à l'ordre 2, c'est pour cela que Magnus Simpson est d'ordre 2.

En comparant, chacune des méthodes, il semble que la méthode Magnus avec la méthode de Simpson pour approximer Ω_2 soit la plus précise pour notre cas test. En effet, elle fait partie des méthodes avec l'ordre le plus élevé (2) et diffère d'une demie puissance comparée à celle de Magnus Point-Milieu (la courbe Magnus Simpson sur la figure 5 est en dessous de Magnus Point Milieu), suggérant sa meilleure précision.

Il est alors intéressant de visualiser sur une sphère la trajectoire des colonnes de notre solution de notre EDO de départ calculée grâce à la méthode Magnus-Simpson.

On rappelle que e_1, e_2 et e_3 sont les vecteurs de la base canonique de $\mathbb{R}^3$. On affiche les 3 colonnes sur la sphère (rouge : première colonne, vert : deuxième colonne et bleu : troisième colonne).

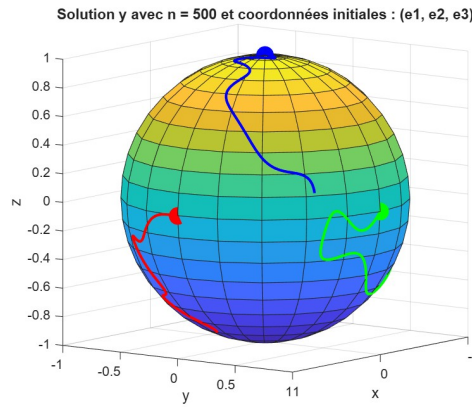


FIGURE 6 – Solution y calculée avec Magnus-Simpson avec $n = 500$ et coordonnées initiales (e_1, e_2, e_3)

4 EDO à coefficients dépendant de la position sur groupe de Lie

Si A dépend de y , nous avons l'équation suivante : $y'(t) = A(t, y(t))y(t)$ que l'on simplifie en $y'(t) = A(y(t))y(t)$.

4.1 Méthodes de Runge-Kutta Munthe-Kaas :

Cette fois-ci, nous ne pouvons plus utiliser la méthode de Magnus puisque nous ne connaissons plus la matrice A : elle dépend de y . Mais on peut écrire la solution y comme une exponentielle ($y = \exp(\Omega(t))y_0$).

Dans ce cas, on se référera à [21, 24, 22, 23] qui définissent les méthodes dites de RKMK :

— définir la solution numérique par $y_{n+1} = \exp(\Omega_1)y_n$.

— considérer l'équation différentielle suivante :

$$\dot{\Omega} = A(\exp(\Omega)y_0) + \sum_{k=1}^q \frac{B_k}{k!} ad_{\Omega}^k A(\exp(\Omega)y_0) \quad (1)$$

avec $\Omega(0) = 0$.

— remplacer y_0 par y_n (pour pouvoir trouver la solution locale) et à appliquer la méthode de Runge-Kutta pour avoir une approximation de Ω .

Rappelons que nous savons que la solution peut s'écrire sous la forme $y(t) = \exp(\Omega(t))y_0$. Comme il n'est pas pratique de travailler avec l'opérateur $d\exp_{\Omega}^{-1}$, nous allons tronquer la série, ce qui nous donne l'équation différentielle précédente 1.

RKMK 2 : Calculer $y_{n+1} = \exp(\Omega_1)y_n$

Require: $n > 0$

Ensure: $y_{n+1} = \exp(\Omega_1)y_n$

$h = \frac{|t_f - t_0|}{N}$

$y \leftarrow y_0$

for $k = 1 \rightarrow N$ **do**

if $|y^T y - I| > \epsilon$ ou $|det(y) - 1| > \epsilon$ **then**

y n'appartient plus à $SO(3)$

break

else

$\Omega_n \leftarrow 0$

$k_1 \leftarrow A(\exp(\Omega_n)y) - \frac{1}{2}(\Omega_n A(\exp(\Omega_n)y) - A(\exp(\Omega_n)y)\Omega_n)$

$\Omega_p = \Omega_n + h k_1$

$k_2 \leftarrow A(\exp(\Omega_p)y) - \frac{1}{2}(\Omega_p A(\exp(\Omega_p)y) - A(\exp(\Omega_p)y)\Omega_p)$

$\Omega_n = \Omega_n + h(\frac{k_1}{2} + \frac{k_2}{2})$

$y \leftarrow \exp(\Omega_n)y$

end if

end for

Théorème 4.1. *Si la méthode de Runge-Kutta est d'ordre p et si l'indice q de la troncature satisfait $q \geq p - 2$, alors la méthode RKMK est d'ordre p [12].*

Exemple 4.2. *Maintenant on définit $y(t)$ comme l'action de la matrice A , donc l'action du groupe $SO(3)$ sur le vecteur y_0 appartenant à S^2 (produit matrice vecteur). Cela simplifiera les calculs et réduira le coût numérique. $y(t) = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} (t)$ avec $y_0 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$*

$$A(y) = \begin{pmatrix} 0 & y_1 & y_2 \\ -y_1 & 0 & y_3 \\ -y_2 & -y_3 & 0 \end{pmatrix} \in \mathfrak{so}_3$$

Remarque 4.3. *La solution exacte de cette EDO est périodique. En effet, comme A est antisymétrique, la norme de solution est conservée, ce qui implique qu'elle se trouve impérativement sur la sphère.*

De plus, comme $y'_1 = y_1y_2 + y_2y_3 = y_2(y_1 + y_3)$ et que $y_0 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$, on a $y_1(0) = 0$. La

solution étant unique d'après le théorème de Cauchy-Lipschitz, on a $y_1(t) = 0$ pour tout t . Le mouvement se trouve donc sur le plan (y_2, y_3) . L'intersection de la sphère et de ce plan, nous donne un grand cercle de la sphère. Comme cet espace est fermé et borné donc compact, la solution est périodique.

4.1.1 RKMK2

Nous allons commencer par la méthode de Runge-Kutta d'ordre 2.

Soit $\dot{\Omega}(t) = f(\Omega(t))$

avec $f(\Omega(t)) = A(\exp(\Omega(t))y_n - \frac{1}{2}[\Omega(t), A(\exp(\Omega(t))y_n)])$

$$\Omega_{n+1} = \Omega_n + h(\frac{k_1}{2} + \frac{k_2}{2})$$

avec

$$\begin{aligned} k_1 &= f(\Omega_n) \\ k_2 &= f(\Omega_n + hk_1) \end{aligned}$$

D'après le théorème 4.1 et l'expression de $\dot{\Omega}$, nous allons tronquer à $q=1$.

$$\dot{\Omega} = A(\exp(\Omega)y_n) - \frac{1}{2}[\Omega, A(\exp(\Omega)y_n)]$$

Donc

$$\Omega_{n+1} = \Omega_n + h(\frac{k_1}{2} + \frac{k_2}{2})$$

avec

$$\begin{aligned} k_1 &= A(\exp(\Omega_n)y_n) - \frac{1}{2}[\Omega_n, A(\exp(\Omega_n)y_n)] \\ k_2 &= A(\exp(\Omega_n + hk_1)y_n) - \frac{1}{2}[\Omega_n + hk_1, A(\exp(\Omega_n + hk_1)y_n)] \end{aligned}$$

Après implémentation, la méthode respecte la structure géométrique puisque la matrice appartient toujours à SO_3 . La méthode est d'ordre 2.

Puisque nous avons : $\log_{10}(Err) = -1.99 \cdot \log_{10}(n) + 0.22$, nous avons $Err = n^{-1.99} \cdot 10^{0.22}$ donc $Err \approx \frac{1}{n^2} \cdot C$.

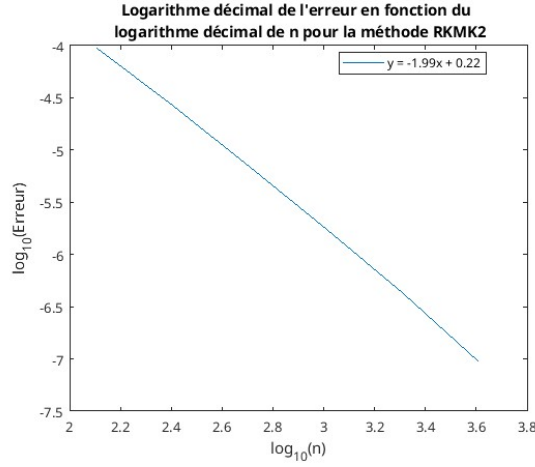


FIGURE 7 – Courbe de convergence de l'erreur de la méthode RKMK2

4.1.2 RKMK4 :

Nous allons poursuivre avec la méthode de Runge-Kutta d'ordre 4.

Soit $\dot{\Omega}(t) = f(\Omega(t))$ avec

$$f(\Omega(t)) = A(\exp(\Omega(t))y_n - \frac{1}{2}[\Omega(t), A(\exp(\Omega(t))y_n)] \\ + \frac{1}{12}[\Omega(t), [\Omega(t), A(\exp(\Omega(t))y_n)]]).$$

avec

$$\begin{aligned} k_1 &= f(\Omega_n) \\ k_2 &= f(\Omega_n + \frac{1}{2}hk_1) \\ k_3 &= f(\Omega_n + \frac{1}{2}hk_2) \\ k_4 &= f(\Omega_n + hk_3) \end{aligned}$$

D'après le théorème 4.1 et l'expression de $\dot{\Omega}$, nous allons tronquer à q=2.

$$\dot{\Omega} = A(\exp(\Omega)y_n) - \frac{1}{2}[\Omega, A(\exp(\Omega)y_n)] + \frac{1}{12}[\Omega, [\Omega, A(\exp(\Omega)Y_n)]]$$

Donc

$$\Omega_{n+1} = \Omega_n + h(\frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6})$$

avec

$$\begin{aligned}
k_1 &= A(\exp(\Omega_n)y_n) - \frac{1}{2}[\Omega_n, A(\exp(\Omega_n)y_n)] + \frac{1}{12}[\Omega_n, [\Omega_n, A(\exp(\Omega_n)y_n)]] \\
k_2 &= A(\exp(\Omega_n + \frac{1}{2}hk_1)y_n) - \frac{1}{2}[\Omega_n + \frac{1}{2}hk_1, A(\exp(\Omega_n + \frac{1}{2}hk_1)y_n)] + \\
&\quad \frac{1}{12}[\Omega_n + \frac{1}{2}hk_1, [\Omega_n + \frac{1}{2}hk_1, A(\exp(\Omega_n + \frac{1}{2}hk_1)y_n)]] \\
k_3 &= A(\exp(\Omega_n + \frac{1}{2}hk_2)y_n) - \frac{1}{2}[\Omega_n + \frac{1}{2}hk_2, A(\exp(\Omega_n + \frac{1}{2}hk_2)y_n)] + \\
&\quad \frac{1}{12}[\Omega_n + \frac{1}{2}hk_2, [\Omega_n + \frac{1}{2}hk_2, A(\exp(\Omega_n + \frac{1}{2}hk_2)y_n)]] \\
k_4 &= A(\exp(\Omega_n + hk_3)y_n) - \frac{1}{2}[\Omega_n + hk_3, A(\exp(\Omega_n + hk_3)y_n)] + \\
&\quad \frac{1}{12}[\Omega_n + hk_3, [\Omega_n + hk_3, A(\exp(\Omega_n + hk_3)y_n)]]
\end{aligned}$$

Après implémentation, la méthode respecte bien la structure géométrique du problème, la matrice appartient toujours à $SO(3)$. La méthode est d'ordre 4.

Puisque nous avons : $\log_{10}(Err) = -3.97 \cdot \log_{10}(n) + 0.82$, nous avons $Err = n^{-3.97} \cdot 10^{0.82}$ donc $Err \approx \frac{1}{n^4} \cdot C$.

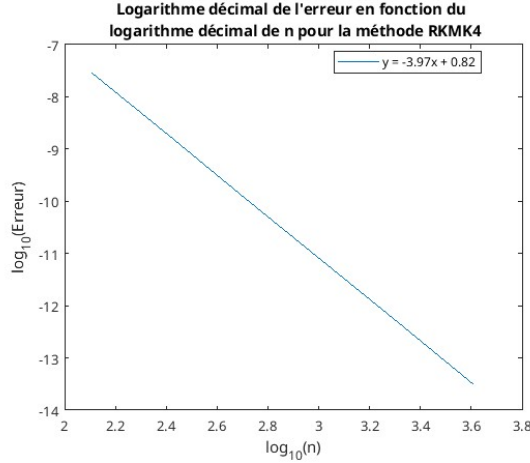


FIGURE 8 – Courbe de convergence de l'erreur de la méthode RKMK4

Pour autant, il est important de bien initialiser nos paramètres d'entrée. Effectivement, avec un temps final égal à 200, ainsi qu'une subdivision de 300 intervalles, les paramètres du schéma ne sont pas bien posés, les pas de temps sont trop grands. Il ne pourra pas être précis comme nous pouvons le voir sur la figure 9.

Solution y avec $n = 300$ $t_f = 200$ et coordonnées initiales : $(0, 0, 1)$

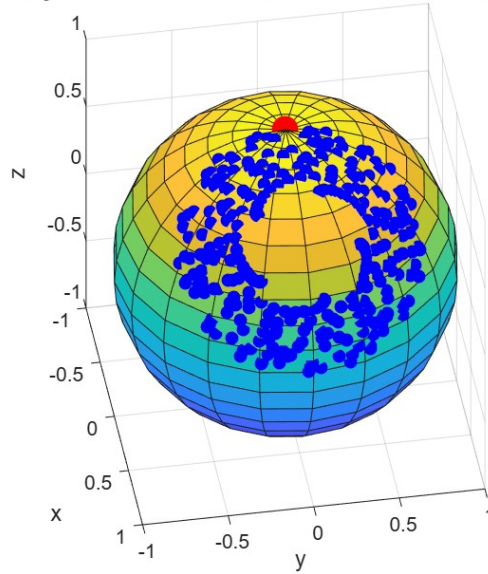


FIGURE 9 – Solution y sur S_2 avec $t_f = 200$

En optimisant bien le nombre d'intervalles et le temps final, on peut retrouver de meilleurs résultats.

Voici une représentation de la solution à deux temps finaux différents :

Solution y avec $n = 500$ $t_f = 2$ et coordonnées initiales : $(0, 0, 1)$

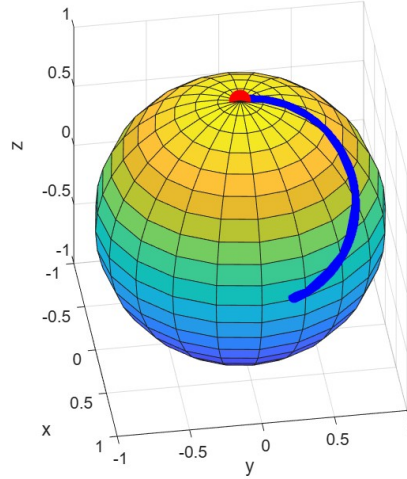


FIGURE 10 – Solution y sur S_2 avec $t_f = 2$

Solution y avec $n = 500$ $t_f = 5$ et coordonnées initiales : $(0, 0, 1)$

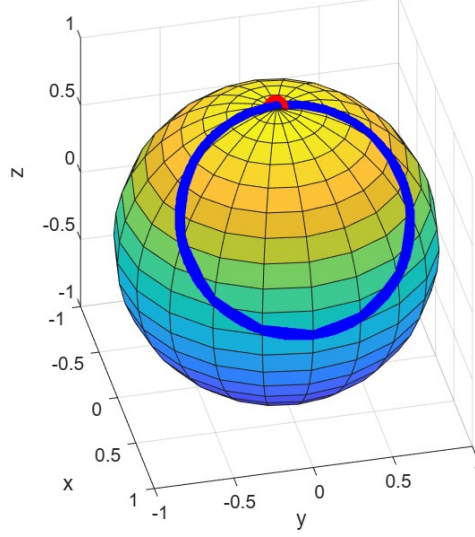


FIGURE 11 – Solution y sur S_2 avec $t_f = 5$

Cette fois-ci sur la figure 11, on remarque que la solution calculée semble converger vers une solution bornée. Bien qu'une méthode numérique ne puisse généralement pas restituer une solution parfaitement périodique à cause de l'accumulation des erreurs de troncature, le comportement obtenu reste particulièrement cohérent.

En résumé, nos deux méthodes RKMK s'avèrent très performantes pour respecter la structure géométrique intrinsèque du problème. Leur propriété fondamentale est que, même avec un pas de temps grossier, la solution approchée reste ancrée sur la sphère. L'ajustement et le raffinement du pas de temps permettent ensuite de gagner en précision géométrique sur le tracé de la trajectoire elle-même, confirmant un ordre de convergence de 2 pour RKMK2 et de 4 pour RKMK4. Ainsi, le choix des paramètres de simulation permet de garder à la fois la précision du mouvement et la robustesse géométrique de la méthode.

4.2 Méthodes sans commutateurs :

Depuis le début, nos constructions reposent sur l'opérateur adjoint $[\Omega, A] = \Omega A - A \Omega$. Cependant, l'augmentation de l'ordre de convergence des méthodes RKMK nécessite d'évaluer des termes de plus en plus élevés dans le développement de la série de Baker. Cela se traduit par une accumulation de commutateurs imbriqués, dont la complexité algébrique et le coût de calcul deviennent rapidement prohibitifs à implémenter.

Pour éviter cette lourdeur algorithmique et s'affranchir du calcul explicite de ces commutateurs tout en conservant les propriétés géométriques (comme le maintien sur la sphère), plusieurs alternatives ont été développées. On peut notamment citer les

méthodes de Crouch et Grossman [9], les approches d'Owren et Marthinsen sur une utilisation de ces méthodes sur des arbres de Lie [28], ou encore les techniques d'intégration de type Commutator-Free introduites par Celledoni [7] et approfondies par Owren [27].

4.2.1 Crouch-Grossman à 3 étapes

Crouch-Grossman : Calculer $y_{n+1} = \exp(b_s K_s) \dots \exp(b_1 K_1)$

Require: $n > 0$

Ensure: $y_{n+1} = \exp(b_s K_s) \dots \exp(b_1 K_1)$

$$h = \frac{|t_f - t_0|}{N}$$

$y \leftarrow y_0$

initialisation des $a_{i,j}$ et b_j

for $k = 1 \rightarrow N$ **do**

if $|y^T y - I| > \epsilon$ ou $|\det(y) - 1| > \epsilon$ **then**

y n'appartient plus à $SO(3)$

 break

else

$$K_1 \leftarrow hA(y)$$

$$Y_2 = \exp(a_{21} K_1)$$

$$K_2 = hA(Y_2)$$

$$Y_3 = \exp(a_{32} K_2) \exp(a_{31} K_1) y$$

$$K_3 = hA(Y_3)$$

$$y \leftarrow \exp(b_3 K_3) \exp(b_2 K_2) \exp(b_1 K_1) y$$

end if

end for

avec les coefficients :

- $a_{ij\alpha}$ réels
- $b_{\alpha j}$ réels
- s est le nombre d'étapes.
- $i, j = 1, \dots, s$

Crouch et Grossman ont présenté plusieurs solutions du système de conditions à vérifier, deux d'entre elles sont [7] :

0				0			
-1/24	-1/24			3/4	3/4		
17/24	161/24	-6		17/24	119/216	17/108	
	1	-2/3	2/3		13/51	-2/3	24/17

Ce tableau est représenté sous forme de "Tableau Butcher" :

c_1	a_{11}	a_{12}	a_{13}
c_2	a_{21}	a_{22}	a_{23}
c_3	a_{31}	a_{32}	a_{33}
	b_1	b_2	b_3

En utilisant les coefficients du premier tableau, nous avons cela :

$$\begin{aligned}
K_1 &= A(Y^{(1)}) \\
Y^{(2)} &= \exp(ha_{21}K_1)Y_n \\
K_2 &= A(Y^{(2)}) \\
Y^{(3)} &= \exp(ha_{32}K_2) \cdot \exp(ha_{31}K_1)Y_n \\
K_3 &= A(Y^{(3)}) \\
Y^{n+1} &= \exp(hb_3K_3) \cdot \exp(hb_2K_2) \cdot \exp(hb_1K_1)Y_n
\end{aligned}$$

Après implémentation, la matrice appartient toujours à $SO(3)$. La méthode est d'ordre 3.

Puisque nous avons : $\log_{10}(Err) = -3.01 \cdot \log_{10}(n) + 0.64$, nous avons $Err = n^{-3.01} \cdot 10^{0.64}$ donc $Err \approx \frac{1}{n^3} \cdot C$.

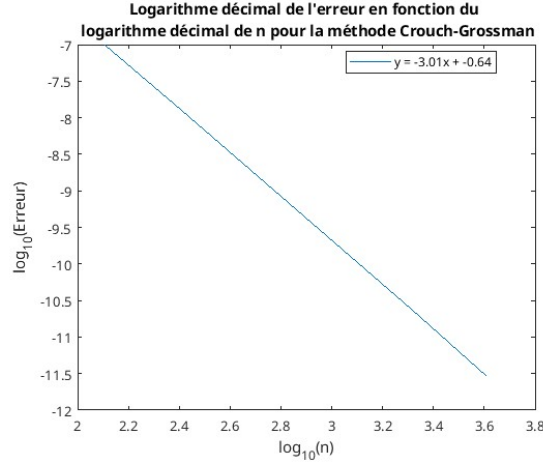


FIGURE 12 – Courbe de convergence de l'erreur de la méthode Crouch-Grossman

En faisant notre méthode à l'ordre 3, nous utilisons 6 exponentielles de matrice. il est alors intéressant de voir le nombre d'utilisations d'exponentielles par ordre. Pour un ordre 4, il est nécessaire d'utiliser 15 exponentielles car il y a 5 étapes, dûes aux nombres de conditions sur les coefficients. Pour un ordre 5, il est difficile de savoir exactement le nombre d'étapes, une approximation de 9 étapes a été faite par comparaison du nombre de conditions. Cela revient donc à 45 exponentielles d'utilisées pour l'ordre 5 [12].

4.2.2 Méthodes Commutator-Free (CF)

Bien que les méthodes de Crouch-Grossman s'affranchissent des commutateurs, elles nécessitent l'évaluation d'un grand nombre d'exponentielles de matrices à chaque étape. Le coût d'évaluation d'une seule exponentielle de matrice de taille $n \times n$ est typiquement de l'ordre de $25n^3$ opérations, tandis qu'un commutateur classique ne requiert que $4n^3$ opérations lorsque aucune structure d'algèbre de Lie n'est exploitée [7]. Les méthodes

RKMK, bien que plus complexes à implémenter à cause des commutateurs imbriqués de l'opérateur dexinv, s'avèrent donc souvent moins coûteuses par étape qu'une méthode de Crouch-Grossman classique.

Pour concilier le fait de ne pas utiliser de commutateurs sans faire exploser le nombre d'exponentielles à évaluer, Celledoni [7] a introduit les méthodes Commutator-Free (CF). L'idée centrale est de remplacer le produit d'exponentielles multiples des méthodes de Crouch-Grossman par un produit réduit d'exponentielles de combinaisons linéaires bien choisies des matrices de direction.

La formule de Baker-Campbell-Hausdorff (BCH) joue un rôle clé dans l'analyse de ces méthodes, non pas lors de leur implémentation numérique, mais pour établir leurs conditions d'ordre théoriques.

Théorème 4.4 (Formule de BCH). [2] *Soient X et Y deux matrices de l'algèbre de Lie. Alors :*

$$\exp(X)\exp(Y) = \exp\left(X + Y + \frac{1}{2}[X, Y] + \frac{1}{12}([X, [X, Y]] - [Y, [X, Y]]) + \dots\right)$$

Lors de la conception d'un schéma CF, la formule de BCH permet de développer théoriquement le produit d'exponentielles sous la forme d'une unique exponentielle contenant une série de commutateurs. En simplifiant et en identifiant les conditions d'ordre sur le papier, on détermine les coefficients du schéma (comme ceux des méthodes CF3 et CF4) permettant d'annuler les termes d'erreur jusqu'au degré souhaité, sans avoir à manipuler explicitement de crochets de Lie dans l'algorithme.

Une fois ces coefficients fixés sur le papier, l'algorithme numérique final n'implémente aucun commutateur : il se contente d'évaluer le produit direct des exponentielles, optimisant ainsi significativement le temps de calcul par rapport aux méthodes RKMK d'ordre équivalent.

Introduisons un nouvel algorithme de méthodes sur groupe de Lie :

$$\begin{aligned} Y^{(i)} &= \exp(h(a_{i,1,J}K_1 + a_{i,2,J}K_2 + \dots + a_{i,i-1,J}K_{i-1})) \\ &\quad \dots \exp(h(a_{i,1,1}K_1 + a_{i,2,1}K_2 + \dots + a_{i,i-1,1}K_{i-1}))Y_n, \\ K_i &= A(Y^{(i)}), \end{aligned}$$

$$\begin{aligned} Y_{n+1} &= \exp(h(b_{J1}K_1 + \dots + b_{Js}K_s)) \\ &\quad \dots \exp(h(b_{11}K_1 + \dots + b_{1s}K_s))Y_n. \end{aligned}$$

avec les coefficients :

- $a_{ij\alpha}$ réels
- $b_{\alpha j}$ réels
- J compte le nombre d'exponentielles à chaque étape
- p est l'ordre de la méthode
- s est le nombre d'étapes.
- $i, j = 1, \dots, s$

— $\alpha = 1, \dots, J$

CF_p : Calculer $CF_p : y_{n+1} = \exp(\sum_k b_{Jk} K_k) \dots \exp(\sum_k b_{1k} K_k)$

Require: $n > 0$

Ensure: $y_{n+1} = \exp(\sum_k b_{Jk} K_k) \dots \exp(\sum_k b_{1k} K_k)$

$$h = \frac{|t_f - t_0|}{N}$$

$y \leftarrow y_0$

initialisation des $a_{i,j}$ et b_j

for $k = 1 \rightarrow N$ **do**

if $|y^T y - I| > \epsilon$ ou $|det(y) - 1| > \epsilon$ **then**

y n'appartient plus à $SO(3)$

 break

else

$$y_r \leftarrow \exp(\sum_k a_{r,k,J} K_k) \dots \exp(\sum_k a_{r,k,1} K_k)$$

$$K_r \leftarrow hA(y_r)$$

$$y_{n+1} \leftarrow \exp(\sum_k b_{Jk} K_k) \dots \exp(\sum_k b_{1k} K_k)$$

end if

end for

Dès lors que l'on arrive à trouver les bons coefficients, il est possible de minimiser l'utilisation des exponentielles et d'annuler les commutateurs. Le tableau de Butcher suivant [7] nous donne les bons coefficients :

0			
1/3	1/3		
2/3	0	2/3	
	1/3	0	0
	-1/12	0	3/4

La première ligne en dessous de la ligne horizontale contient les coefficients b_{1j} et la seconde contient b_{2j} . En utilisant les coefficients du premier tableau, nous avons cela :

$$K_1 = hA(Y^{(1)})$$

$$Y^{(2)} = \exp(a_{211} K_1) Y_n$$

$$K_2 = hA(Y^{(2)})$$

$$Y^{(3)} = \exp(a_{311} K_1 + a_{321} K_2) Y_n$$

$$K_3 = hA(Y^{(3)})$$

$$Y^{n+1} = \exp(b_{21} K_1 + b_{22} K_2 + b_{23} K_3) \cdot \exp(b_{11} K_1 + b_{12} K_2 + b_{13} K_3) Y_n$$

Avec ces coefficients, la méthode est plus connue sous le nom de : Runge Kutta d'ordre 3 de Heun, plus souvent appelée : CF3.

Après implémentation, la méthode respecte la structure géométrique, la matrice appartient toujours à SO_3 . La méthode est d'ordre 3.

Puisque nous avons : $\log_{10}(Err) = -3.00 \cdot \log_{10}(n) + 0.68$, nous avons $Err = n^{-3.00} \cdot 10^{0.68}$ donc $Err \approx \frac{1}{n^3} \cdot C$.

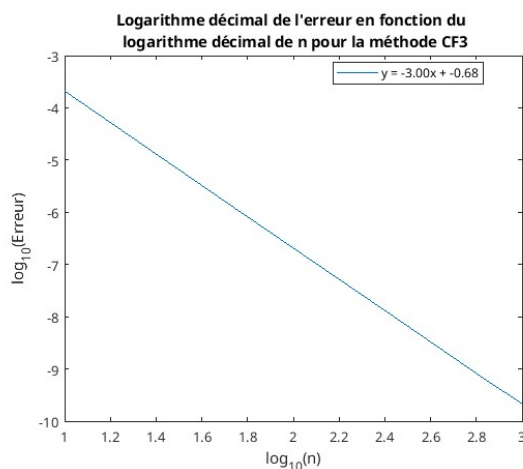


FIGURE 13 – Courbe de convergence de l'erreur de la méthode CF3

A l'ordre 3, on remarque nous avons baissé le nombre d'exponentielles utilisées, passant de 6 à 5.

De la même manière que dans CF3, à l'ordre 4, il est possible de trouver de bons coefficients pour minimiser le nombre d'exponentielles et de commutateurs, ce sont les coefficients de la méthode de Runge-Kutta 4.

Le tableau de Butcher suivant nous donne les bons coefficients :

0				
1/2	1/2			
1/2	0	1/2		
1	1/2	0	0	
	-1/2	0	1	
	1/4	1/6	1/6	-1/12
	-1/12	1/6	1/6	1/4

Il faut lire le tableau de cette manière :

0				
c_2	a_{211}			
c_3	a_{311}	a_{321}		
c_4	a_{411}	a_{421}	a_{431}	
	a_{412}	a_{422}	a_{432}	
	b_{11}	b_{12}	b_{13}	b_{14}
	b_{21}	b_{22}	b_{23}	b_{24}

La première ligne en dessous de la ligne horizontale contient les coefficients b_{1j} et la

seconde contient b_{2j} . En utilisant les coefficients du premier tableau, nous avons cela :

$$\begin{aligned}
K_1 &= hA(Y^{(1)}) \\
Y^{(2)} &= \exp(a_{211}K_1)Y_n \\
K_2 &= hA(Y^{(2)}) \\
Y^{(3)} &= \exp(a_{311}K_1 + a_{321}K_2)Y_n \\
K_3 &= hA(Y^{(3)}) \\
Y^{(4)} &= \exp(a_{411}K_1 + a_{421}K_2 + a_{431}K_3) \cdot \exp(a_{412}K_1 + a_{422}K_2 + a_{432}K_3) \\
K_4 &= hA(Y^{(4)}) \\
Y^{n+1} &= \exp(b_{21}K_1 + b_{22}K_2 + b_{23}K_3 + b_{24}K_4) \cdot \exp(b_{11}K_1 + b_{12}K_2 + b_{13}K_3 + b_{14}K_4)Y_n
\end{aligned}$$

Avec ces coefficients, la méthode se réfère à la méthode de Runge-Kutta d'ordre 4 et elle est appelée : CF4.

Après implémentation, la méthode fonctionne bien, la matrice appartient toujours à $SO(3)$. La méthode est d'ordre 4. Puisque nous avons : $\log_{10}(Err) = -4.07 \cdot \log_{10}(n) + 0.52$, nous avons $Err = n^{-4.07} \cdot 10^{0.52}$ donc $Err \approx \frac{1}{n^4} \cdot C$.

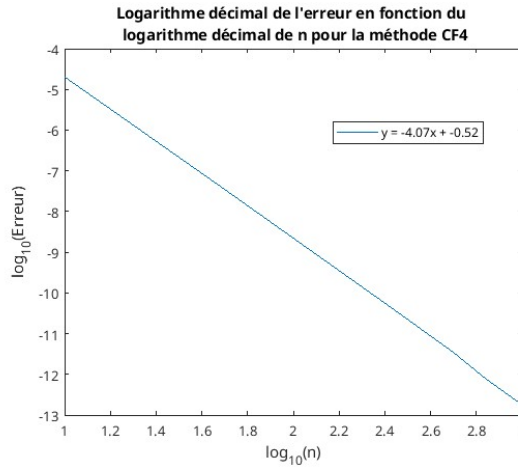


FIGURE 14 – Courbe de convergence de l'erreur de la méthode CF4

Cette fois-ci, nous passons de 15 exponentielles à 6 exponentielles.

En mettant un temps final à 200 et une subdivision de 300 intervalles, nous obtenons ce graphe suivant.

Solution y avec $n = 300$ $t_f = 200$ et coordonnées initiales : $(0, 0, 1)$

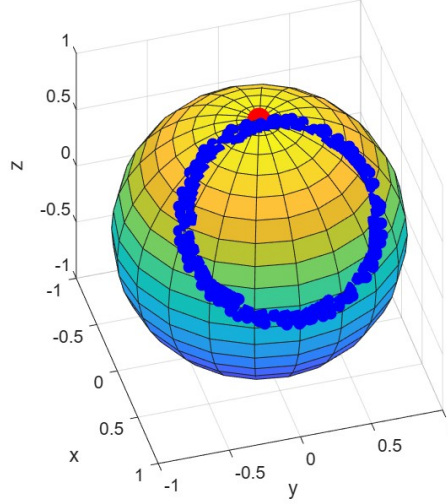


FIGURE 15 – Solution y à $t_f = 200/n = 300$ avec la méthode CF4

Comparée à la figure 9, la solution de la figure 14 semble plus précise, puisqu'elle paraît converger plus précisément vers une solution périodique.

Avec $n = 500$ et un temps final à 5, nous obtenons le graphe suivant.

Solution y avec $n = 500$ $t_f = 5$ et coordonnées initiales : $(0, 0, 1)$

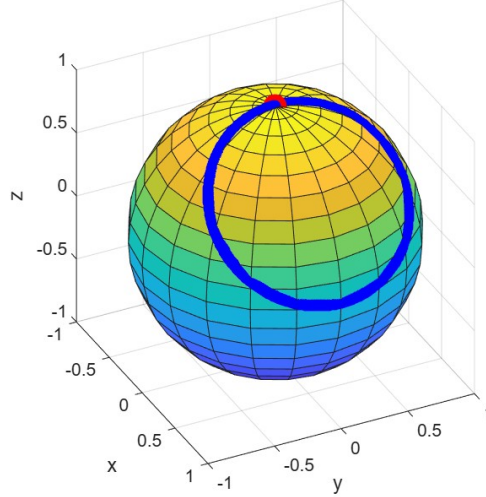


FIGURE 16 – Solution y à $t_f = 5/n = 500$ avec la méthode CF4

Comme sur la figure 12, la solution calculée sur la figure 16 semble converger vers une solution périodique.

La méthode CF4 semble la plus judicieuse pour notre cas test. Elle est d'ordre 4, est plus simple à implémenter, nécessite moins de temps de calculs et est plus précise que

les précédentes. Cette dernière a permis de réduire de manière significative le coût informatique global en minimisant le nombre d'exponentielles de matrices requises (passant notamment de 15 à 6 exponentielles).

Il est intéressant de comparer le temps de calcul entre les deux méthodes RKMK4 et CF4. Pour cela on pose $t_f = 5$ et on fait varier $N = [100, 500, 1000, 5000, 10000, 50000, 100000]$ et on fait le graphe du logarithme décimal du temps d'exécution en fonction du logarithme décimal de N .

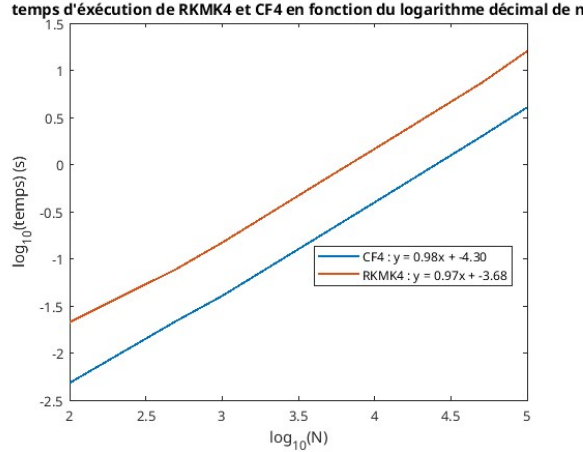


FIGURE 17 – Logarithme décimal du temps en fonction du logarithme décimal de N

Sur la figure 17, observe que les temps d'exécution des deux méthodes augmentent de façon quasi linéaire avec le nombre d'intervalles, comme en témoignent les pentes de régressions log-log très proches de 1.

Néanmoins, la méthode CF4 se révèle plus performante que RKMK4. Bien que les deux algorithmes partagent la même complexité temporelle, leurs ordonnées à l'origine montre une différence d'un demi-ordre de grandeur. La méthode CF4 s'exécute environ 4 fois plus vite que la méthode RKMK4 sur l'ensemble de N testés.

Pour autant, il est à noter que nous travaillons en dimension 3, le calcul d'exponentielle et de commutateurs prend un temps minime. Il serait donc judicieux d'étudier ce problème sur des dimensions plus élevées et des méthodes d'ordre plus élevé, pour pouvoir conclure à une plus grande différence de performance.

5 Conclusion

Ce stage, consacré à l'étude et au développement de méthodes numériques géométriques sur les groupes de Lie, a permis de répondre à la problématique de la simulation de déplacements directement sur la sphère S^2 . L'objectif était de concevoir des algorithmes capables de préserver la structure géométrique du groupe sur lequel nous travaillons, évitant donc les instabilités numériques des méthodes d'intégration classiques.

Dans un premier temps, nous avons pu montrer pourquoi nous pouvions utiliser le groupe de rotations $SO(3)$ et son algèbre de Lie associée $\mathfrak{so}(3)$ pour simuler S^2 . Les deux espaces sont reliés par l'exponentielle de matrices. Cette démarche s'est avérée indispensable pour contraindre naturellement les trajectoires sur la sphère.

La partie d'implémentation numérique sous MATLAB a permis de confronter et de valider l'ordre de convergence théorique de plusieurs types de méthodes :

- Les méthodes d'Euler et Runge-Kutta 4 ont mis en avant leur défaillance immédiate (Euler Explicite qui quitte le groupe $SO(3)$ dès la première itération et RK4 en 5 itérations), tandis que la méthode de Lie-Euler a validé un ordre 1 tout en respectant la structure du groupe.
- Les méthodes de Magnus ont permis d'atteindre des ordres de convergence plus élevés (ordre 1 pour Rectangle à gauche et ordre 2 pour le Point-Milieu et Simpson).
- Les méthodes de Runge-Kutta Munthe-Kaas (RKMK2 et RKMK4) ont démontré leur efficacité lorsque l'opérateur dépend de la position, confirmant expérimentalement les ordres de convergences attendus (2 et 4).

Enfin, une partie du projet a été consacrée à l'optimisation des algorithmes par l'étude des méthodes sans commutateurs, telles que Crouch-Grossman et les méthodes Commutator-Free (CF3 et CF4). Celles-ci réduisent significativement le coût de calcul en diminuant le nombre d'exponentielles de matrices à évaluer : pour un schéma d'ordre 4, on passe notamment de 15 à 6 exponentielles. Pour une analyse comparative plus poussée et des détails numériques approfondis sur l'efficacité de ces intégrateurs, on pourra se référer à la revue d'Iserles. [13].

Finalement, il serait intéressant d'étudier ces méthodes sur des ordres plus élevés pour obtenir une différence de temps de calcul encore plus significative. De plus, au-delà de l'équation test traitée dans ce rapport, le groupe $SO(3)$ joue un rôle fondamental en ingénierie aérospatiale et en robotique car il modélise les matrices d'orientation d'un corps rigide dans $\mathbb{R}^3$. Les méthodes géométriques comme CF4 et RKMK4 trouvent une application directe dans le contrôle de trajectoire de drones ou de satellites.

À plus long terme, il serait pertinent d'étendre ce cadre géométrique aux équations différentielles stochastiques (EDS) afin de modéliser des systèmes soumis à des perturbations aléatoires, en s'appuyant sur les schémas d'ordre supérieur récemment développés [4, 5]. D'autre part, la solidité de ces intégrateurs repose sur des structures algébriques sous-jacentes rigoureuses — notamment les algèbres de pré-Lie et les séries de Post-Lie — dont l'étude fondamentale ouvre la voie à des généralisations sur des variétés encore plus complexes [26, 25, 6].

Références

- [1] T. Adamski. Notes de cours de géométrie algébrique linéaire. <https://perso.eleves.ens-rennes.fr/people/teofil.adamski/notes-cours/GAL.pdf>, 2023.

- [2] M. J. H. Al-Kaabi, K. Ebrahimi-Fard, D. Manchon, et al. Post-Lie Magnus expansion and BCH-recursion. *SIGMA. Symmetry, Integrability and Geometry : Methods and Applications*, 18 :023, 2022.
- [3] S. Blanes, F. Casas, J. Oteo, and J. Ros. Magnus and fer expansions for matrix differential equations : the convergence problem. *Journal of Physics A : Mathematical and General*, 31(1) :259, 1998.
- [4] E. Bronasco, A. Busnot Laurent, and B. Huguet. High order integration of stochastic dynamics on Riemannian manifolds with frozen-flow methods. *arXiv :2503.21855*, 2025.
- [5] A. Busnot Laurent and S. Macé. Characterization of frozen-flow methods for the long time sampling of ergodic dynamics on Riemannian manifolds. *Preprint*, 2026.
- [6] A. Busnot Laurent, H. Munthe-Kaas, and G. S. Venkatesh. The free tracial post-Lie-Rinehart algebra of planar aromatic trees for the numerical preservation of volume forms by Lie-group methods. *Preprint*, 2026.
- [7] E. Celledoni, A. Marthinsen, and B. Owren. Commutator-free Lie group methods. *Future Generation Computer Systems*, 19(3) :341–352, 2003.
- [8] R. Cofré and B. Cessac. Dynamics and spike trains statistics in conductance-based integrate-and-fire neural networks with chemical and electric synapses. *Chaos, Solitons & Fractals*, 50 :13–31, 2013. Functionality and Dynamics in Biological Systems.
- [9] P. E. Crouch and R. Grossman. Numerical integration of ordinary differential equations on manifolds. *Journal of Nonlinear Science*, 3 :1–33, 1993.
- [10] M. Erdoğan and M. Özdemir. On exponential of split quaternionic matrices. *Applied Mathematics and Computation*, 315 :468–476, 2017.
- [11] E. Hairer. Solving differential equations on manifolds. *Lecture notes*, 2011.
- [12] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2006. Structure-preserving algorithms for ordinary differential equations.
- [13] A. Iserles, H. Z. Munthe-Kaas, S. P. Nørsett, and A. Zanna. Lie-group methods. In *Acta numerica, 2000*, volume 9 of *Acta Numer.*, pages 215–365. Cambridge Univ. Press, Cambridge, 2000.
- [14] B. Keller. Introduction aux groupes et algèbres de lie. Notes de cours, Université Paris Diderot.
- [15] E. Kikianty and G. Sinnamon. Angular equivalence of normed spaces. *Journal of Mathematical Analysis and Applications*, 454(2) :942–960, 2017.
- [16] S. J. Kilmer. Chapter 7 - homeomorphisms. In *An Introduction to Point-Set Topology*, pages 59–65. Academic Press, 2027.
- [17] J. Lafontaine. Introduction aux variétés différentielles : Nouvelle édition. In *Introduction aux variétés différentielles*. EDP sciences, 2021.
- [18] G. Lindfield and J. Penny. 5 - solution of differential equations. In *Numerical Methods (Third Edition)*, pages 233–281. Academic Press, Boston, third edition edition, 2012.

- [19] W. Magnus. On the exponential solution of differential equations for a linear operator. *Communications on pure and applied mathematics*, 7(4) :649–673, 1954.
- [20] M. Muniz, M. Ehrhardt, M. Günther, and R. Winkler. Strong stochastic runge-kutta–munthe-kaas methods for nonlinear itô sdes on manifolds. *Applied Numerical Mathematics*, 193 :196–203, 2023.
- [21] H. Munthe-Kaas. Lie-Butcher theory for Runge-Kutta methods. *BIT Numer. Math.*, 35 :572–587, 1995.
- [22] H. Munthe-Kaas. Runge-Kutta methods on Lie groups. *BIT Numer. Math.*, 38 :92–111, 1998.
- [23] H. Munthe-Kaas. High order Runge-Kutta methods on manifolds. *Applied Numerical Mathematics*, 29(1) :115–127, 1999.
- [24] H. Munthe-Kaas and A. Zanna. Numerical integration of differential equations on homogeneous manifolds. In *Foundations of Computational Mathematics : Selected Papers of a Conference Held at Rio de Janeiro, January 1997*, pages 305–315. Springer, 1997.
- [25] H. Z. Munthe-Kaas and A. Lundervold. On post-Lie algebras, Lie–Butcher series and moving frames. *Foundations of Computational Mathematics*, 13 :583–613, 2013.
- [26] H. Z. Munthe-Kaas and W. M. Wright. On the Hopf algebraic structure of Lie group integrators. *Found. Comput. Math.*, 8(2) :227–257, 2008.
- [27] B. Owren. Order conditions for commutator-free Lie group methods. *Journal of Physics A : Mathematical and General*, 39(19) :5585, 2006.
- [28] B. Owren and A. Marthinsen. Runge-Kutta methods adapted to manifolds and based on rigid frames. *BIT Numer. Math.*, 39(1) :116–142, 1999.
- [29] R. Rodrigues. Padé approximation and partition function zeros. *Computer Physics Communications*, 326 :110234, 2026.
- [30] J. J. Rotman. *An introduction to the theory of groups*. Springer Science & Business Media, 2012.
- [31] Seewoo. s^2 is not a lie group. https://seewoo5.github.io/math-notes/geometry/S2_is_not_Lie_group/S2notLiegrp.pdf, 2026. Math Notes.
- [32] S. Sommer, T. Fletcher, and X. Pennec. 1 - introduction to differential and riemannian geometry. In X. Pennec, S. Sommer, and T. Fletcher, editors, *Riemannian Geometric Statistics in Medical Image Analysis*, pages 3–37. Academic Press, 2020.
- [33] X. Wang, K. Deng, Z. Peng, and C. Yan. New vector transport operators extending a Riemannian CG algorithm to generalized stiefel manifold with low-rank applications. *Journal of Computational and Applied Mathematics*, 451 :116024, 2024.