



Problème inverse pour la tomographie optique, applications à la mesure de turbidité

Rapport de stage Master 2

Abdou WADE

supervised by

Joubine Aghili

Date : 28 août 2025

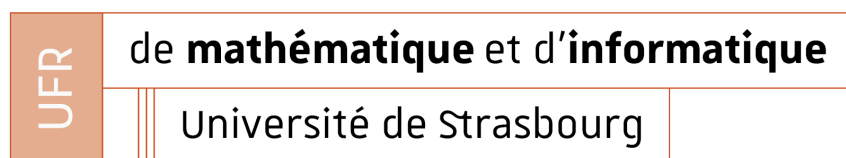


Table des matières

1	Introduction	1
2	Modèle de transfert radiatif	3
2.1	Équation de transfert radiatif (ETR)	3
2.2	Fonction de phase de Henyey–Greenstein	4
2.3	Méthode des ordonnées discrètes (DOM)	6
2.4	Discretisation spatiale et temporelle	6
3	Filtres de Kalman	9
3.1	Introduction	9
3.2	Filtre de Kalman linéaire	9
3.3	Extension au cas non linéaire : filtre de Kalman étendu (EKF)	11
3.4	Interprétation des matrices de covariance et du gain	12
3.5	Application à la tomographie optique	13
4	Problème inverse	16
4.1	Structure du code et approche directe	16
4.2	Application du filtre EKF à la tomographie optique	17
4.3	Analyse et visualisation des résultats	18
4.4	Calcul de la jacobienne du schéma	20
4.5	Estimation des paramètres optiques par EKF	21
5	Résultats numériques	25
5.1	Présentation générale	25
5.2	Analyse détaillée des courbes de convergence	25
5.3	Analyse de configurations extrêmes	30
5.4	Illustrations et interprétations visuelles	31

6 Conclusion and Perspectives	39
7 References	39
Annexe : Interprétation et calcul des incertitudes	41
Annexe : Organisation du code EKF	42
A Calcul formel de la jacobienne du schéma	43

1 Introduction

La tomographie optique est une technique non invasive permettant d’explorer les propriétés internes de milieux diffusants par la mesure de la distribution spatiale et angulaire de la lumière après sa propagation. Les paramètres optiques fondamentaux, tels que les coefficients d’absorption μ_a et de diffusion μ_s , ainsi que le paramètre d’anisotropie g , renseignent sur la composition et la structure du milieu étudié, avec des applications en imagerie biomédicale, surveillance environnementale et sciences des matériaux.

Le modèle de transfert radiatif (ETR) constitue la base de la modélisation physique de la propagation lumineuse dans ces milieux, prenant en compte les phénomènes d’absorption, diffusion et émission locale. Afin de traiter la complexité spatiale et angulaire de l’ETR, la méthode des ordonnées discrètes (DOM) est utilisée pour discrétiser les directions de propagation, tandis qu’un schéma explicite upwind assure la discrétisation spatiale stable et efficace. La diffusion angulaire est décrite par la fonction de phase de Henyey–Greenstein, paramétrée par $g \in [-1, 1]$, qui caractérise la directionnalité du scattering.

L’approche inverse, visant à estimer conjointement le champ lumineux discret et les paramètres optiques internes à partir de mesures lumineuses bruitées, est réalisée via un filtre de Kalman étendu (EKF). Ce filtre permet une estimation récursive et adaptative de l’état augmenté combinant le champ radiatif et les paramètres μ_a, μ_s, g , en assimilant les données expérimentales par confrontation avec un modèle dynamique non linéaire. L’autodifférentiation automatique avec PyTorch pour le calcul exact des jacobiens garantit la cohérence entre la simulation et la linéarisation nécessaire à l’EKF.

Les simulations numériques valident la robustesse de l’approche, démontrant une convergence rapide du coefficient d’absorption μ_a , une correction progressive de la diffusion μ_s , et une estimation plus délicate du paramètre d’anisotropie g , sensible au calibrage des matrices de covariance du modèle (Q) et de l’observation (R). Les analyses incluent également la gestion des cas extrêmes, tels que une forte incertitude initiale ou un bruit important, soulignant l’importance du réglage fin des hyperparamètres pour garantir stabilité et précision.

Enfin, la mise en œuvre modulaire du code en PyTorch, exploitant les calculs GPU et l’autodifférentiation vectorisée, permet de traiter efficacement des états de grandes dimensions et ouvre des perspectives pour l’extension à des modèles plus complexes ou des données expérimentales réelles. Les résultats obtenus constituent un socle pour le développement futur de méthodes avancées en tomographie optique inverse.

Ce rapport est structuré de la manière suivante.

Dans un premier temps, on présente le modèle de transfert radiatif utilisé pour décrire la propagation de la lumière dans un milieu diffusant, incluant les formulations mathématiques, la fonction de phase de Henyey–Greenstein et la méthode numérique des ordonnées discrètes (DOM), ainsi que la discrétisation spatiale et temporelle du modèle.

Ensuite on introduit les filtres de Kalman, en détaillant le filtre linéaire classique et son extension au cas non linéaire, le filtre de Kalman étendu (EKF), adapté à la problématique proposée. Une application spécifique à la tomographie optique est présentée.

La section suivante est consacrée au problème inverse, décrivant l'implémentation numérique, l'utilisation du filtre EKF pour estimer simultanément le champ lumineux et les paramètres optiques, ainsi que l'analyse des résultats obtenus.

Ensuite, la présentation des résultats numériques issus des simulations, incluant des analyses détaillées des courbes de convergence, des scénarios extrêmes, et des illustrations visuelles des estimations.

Des annexes complètent ce document avec des détails techniques, notamment le calcul formel des jacobienues, un modèle paramétrique de démonstration, et une description de l'organisation du code source.

2 Modèle de transfert radiatif

Le modèle de transfert radiatif (ETR) décrit quantitativement la propagation de la lumière dans un milieu absorbant et diffusant. Il permet de représenter l'interaction photon-matière sous la forme d'une équation intégral-différentielle, combinant absorption, diffusion, et émission. L'ETR est largement utilisé en astrophysique, imagerie biomédicale, climatologie ou science des matériaux.

2.1 Équation de transfert radiatif (ETR)

En tomographie optique, nous mesurons la distribution spatiale et angulaire de la lumière après sa propagation dans un milieu diffusant. Cette mesure se fait de manière non invasive. Des propriétés internes du tissu, comme la concentration en oxygène, en eau ou en hémoglobine, peuvent être inférées à partir de ces mesures. Ces propriétés sont caractérisées par les coefficients d'absorption μ_a et de diffusion μ_s , qui représentent respectivement la perte d'énergie lumineuse par absorption (convertie en chaleur ou utilisée dans des réactions photochimiques) et la redirection des photons sans perte d'énergie.

Dans le cadre microscopique, la lumière est décrite comme un flux de photons qui interagissent avec la matière (par absorption, diffusion ou émission). L'énergie d'un photon est donnée par $\Delta E = h\nu = hc/\lambda$, où h est la constante de Planck, ν la fréquence, et λ la longueur d'onde.

Pour modéliser numériquement la propagation lumineuse, nous nous plaçons dans un cadre statistique, en considérant l'intensité radiative $u(t, x, v)$, qui quantifie l'énergie transportée en un point x , suivant une direction v , à un instant t , par unité de surface, de temps et de direction.

L'évolution de u résulte de trois effets : l'extinction (absorption et diffusion hors de la direction v), l'émission locale, et le *scattering* entrant (redirection depuis d'autres angles).

L'équation de transfert radiatif (ETR) s'écrit :

$$\frac{1}{c} \frac{\partial u}{\partial t} + v \cdot \nabla_x u + (\mu_a + \mu_s)u = s + \mu_s \int_{S^1} u \Phi_g(v, v') dv' \quad (1)$$

où c désigne la vitesse de la lumière dans le milieu, s une éventuelle source locale, μ_a et μ_s les coefficients d'absorption et de diffusion du milieu respectivement, et $\Phi_g(v, v')$ la fonction de phase, précisée plus bas.

Les coefficients optiques μ_a (absorption) et μ_s (diffusion) déterminent la probabilité de perte d'énergie et la redistribution angulaire suite aux interactions photon-matière. L'intensité radiative subit ainsi trois effets principaux :

1. **Atténuation locale** : perte d'énergie par absorption et diffusion, modélisée par le terme $(\mu_a + \mu_s)u$.
2. **Transport directionnel** : advection de l'intensité dans la direction v , représentée par $v \cdot \nabla_x u$.

3. **Gain par diffusion** : redirection des photons depuis toutes les directions, pondérée par la fonction de phase $\Phi_g(v, v')$.

Dans la suite, nous nous plaçons dans un régime, monochromatique, négligeant l'émission locale (s constant et isotrope). L'équation se simplifie alors :

$$v \cdot \nabla_x u + (\mu_a + \mu_s)u - \mu_s \int_{S^1} u \Phi_g(v, v') dv' = s \quad (2)$$

Ce modèle est complété par des conditions aux limites (par exemple, gaussienne en entrée sur x , réflexion ou absorption sur les bords, périodicité angulaire sur S^1), adaptées au contexte physique étudié.

2.2 Fonction de phase de Henyey–Greenstein

La diffusion angulaire est modélisée par une fonction de phase Φ_g , qui représente la probabilité pour un photon d'être redirigé de la direction incidente v' vers v . Nous adoptons le modèle de Henyey–Greenstein, proposé initialement dans [1].

En trois dimensions :

$$\Phi_{HG}(v, v') = \frac{1 - g^2}{4\pi (1 + g^2 - 2g v \cdot v')^{3/2}} \quad (3)$$

et, en deux dimensions ($v, v' \in S^1$),

$$\Phi_{HG}^{(2D)}(v, v') = \frac{1 - g^2}{2\pi (1 + g^2 - 2g v \cdot v')} \quad (4)$$

où $g \in [-1, 1]$ est un paramètre d'anisotropie.

La fonction de phase est normalisée, c'est-à-dire : $\int_{S^1} \Phi_{HG}(v, v') dv' = 1$ (ou sur S^2 en 3D). Dans la pratique numérique, nous utilisons une version discrétisée :

$$\Phi_g(v_l, v_k) = \frac{1 - g^2}{4\pi (1 + g^2 - 2g v_l \cdot v_k)^{3/2}},$$

pour des directions discrètes v_l, v_k du maillage angulaire. Nous posons :

$$N(g) = 1 - g^2, \quad D_{l,k}(g) = 1 + g^2 - 2g(v_l \cdot v_k).$$

que nous utilisons pour simplifier les calculs des dérivées dans l'annexe A.

Interprétation de g .

Le paramètre d'anisotropie g mesure la moyenne du cosinus de l'angle de diffusion, soit :

$$g = \langle \cos \theta_k \rangle$$

où θ_k est l'angle entre le rayon incident et dévié sur le k ième pas.

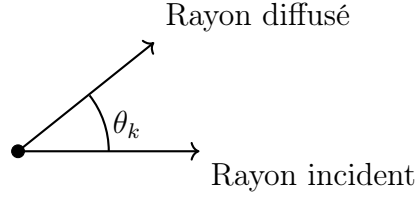


FIGURE 1 – Définition de l'angle de diffusion θ_k entre le rayon incident et le rayon diffusé.

Il contrôle la directionnalité de la diffusion :

- $g = 0$: diffusion isotrope (distribution uniforme),
- $g > 0$: diffusion orientée majoritairement vers l'avant,
- $g < 0$: diffusion principalement rétrograde.

Plus $|g|$ approche 1, plus la diffusion est directionnelle. Cet effet est illustré à la figure 2, où l'on observe l'influence de g sur la concentration angulaire de la distribution.

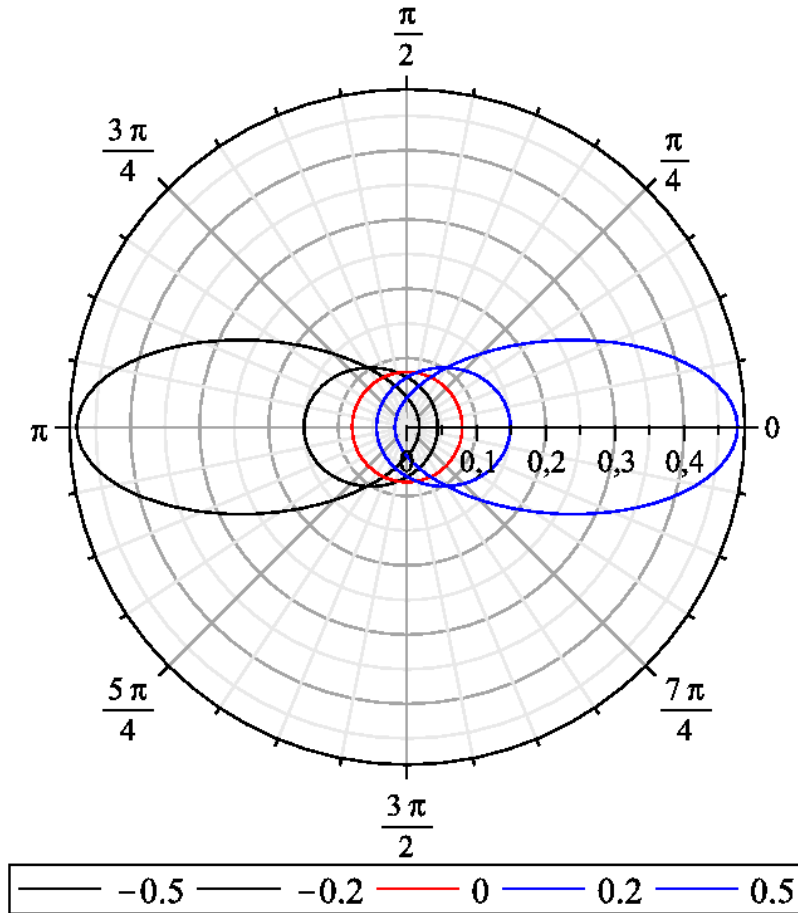


FIGURE 2 – Fonction de phase de Henyey–Greenstein (voir équations 3, 4), Fonction de phase Henyey-Greenstein pour diverses valeurs de g . Le rayonnement incident arrive de la gauche.

La dynamique du paramètre g impacte la nature de la propagation : une forte anisotropie augmente la pénétration de la lumière mais diminue la part rétrodiffusée détectable en surface. À l'inverse, la diffusion isotrope favorise une sortie plus rapide de la lumière. Cette propriété conditionne la sensibilité du problème inverse : un g élevé réduit l'information accessible sur les zones profondes et augmente les corrélations entre μ_a , μ_s et g , ce qui affecte l'identifiabilité des paramètres.

2.3 Méthode des ordonnées discrètes (DOM)

La résolution numérique de l'ETR nécessite d'approcher l'intégrale angulaire par une somme finie. La méthode des ordonnées discrètes (Discrete Ordinates Method, DOM) consiste à discrétiser l'espace des directions en un ensemble fini de directions $\{v_k\}_{k=0}^{K-1}$.

En dimension 2, les directions sont uniformément réparties sur le cercle unité S^1 :

$$\theta_k = \frac{2\pi k}{K}, \quad v_k = (\cos \theta_k, \sin \theta_k), \quad k = 0, \dots, K-1.$$

La mesure angulaire est remplacée par une quadrature numérique, approximant l'intégrale sur S^1 par une somme pondérée :

$$\int_{S^1} f(v) dv \approx \sum_{k=0}^{K-1} w_k f(v_k),$$

avec $w_k = \frac{2\pi}{K}$ les poids de quadrature.

Cette discrétisation transforme l'ETR en un système couplé de K équations scalaires, chaque équation correspondant à une direction v_l :

$$v_l \cdot \nabla_x u + (\mu_a + \mu_s)u - \mu_s \sum_{k=0}^{K-1} w_k u(v_k) \Phi_g(v_l, v_k) = s(v_l).$$

La méthode Discrete Ordinates Method (DOM) présente des avantages comme : une modélisation précise de l'anisotropie via la fonction de phase Φ_g , une parallélisation facile des calculs, et une complexité maîtrisée en $\mathcal{O}(n^2 K)$ avec n la taille du maillage spatial et K le nombre de directions angulaires.

2.4 Discrétisation spatiale et temporelle

Pour la résolution numérique de l'équation de transfert radiatif (ETR), la discrétisation spatiale est réalisée par un schéma explicite par différences finies amont (upwind), adapté à la direction de propagation des photons.

Soit $u_{i,j,l}^n$ l'approximation de l'intensité au pas de temps t^n , au point (x_i, y_j) et dans la direction $v_l = (v_{l,x}, v_{l,y})$. Le schéma explicite est donné par :

$$\begin{aligned}
& \frac{u_{i,j,l}^{n+1} - u_{i,j,l}^n}{\delta t} + \vec{v}_l \cdot \left[\frac{u_{i,j,l}^n - u_{up(i),j,l}^n}{\frac{\Delta x}{u_{i,j,l}^n - u_{i,up(j),l}^n}} \right] + (\mu_a + \mu_s) u_{i,j,l}^n \\
& - \mu_s \sum_{k=0}^{K-1} w_k u_{i,j,k}^n \Phi_g(v_l, v_k) = s(t^n, x_{i,j}, v_l)
\end{aligned} \tag{5}$$

où δt est le pas de temps, Δx et Δy les pas d'espace dans les directions x et y , respectivement, et $up(i)$ et $up(j)$ sont les indices amont pour les directions x et y , et sont définis par :

$$up_x(i, l) = \begin{cases} i-1, & \text{si } v_{l,x} \geq 0 \\ i+1, & \text{si } v_{l,x} < 0 \end{cases} \quad \text{et} \quad up_y(j, l) = \begin{cases} j-1, & \text{si } v_{l,y} \geq 0 \\ j+1, & \text{si } v_{l,y} < 0 \end{cases}$$

respectivement.

Le terme de transport $\vec{v}_l \cdot \nabla u$ est discrétisé par un schéma amont, en tenant compte de la direction de propagation v_l . Pour la composante x , si $v_{l,x} \geq 0$, l'information provient du point x_{i-1} , on approxime donc

$$\frac{\partial u}{\partial x} \approx \frac{u_{i,j,l}^n - u_{i-1,j,l}^n}{\Delta x}.$$

Si $v_{l,x} < 0$, l'information vient de x_{i+1} :

$$\frac{\partial u}{\partial x} \approx \frac{u_{i+1,j,l}^n - u_{i,j,l}^n}{\Delta x}.$$

Dans ce cas, l'approximation devient :

$$v_{l,x} \frac{\partial u}{\partial x} \approx v_{l,x} \frac{u_{i,j,l}^n - u_{up_x(i,l),j,l}^n}{\Delta x}.$$

Par analogie, pour la composante y , on utilise :

$$v_{l,y} \frac{\partial u}{\partial y} \approx v_{l,y} \frac{u_{i,j,l}^n - u_{i,up_y(j,l),l}^n}{\Delta y}.$$

En combinant ces deux contributions, on obtient le terme d'advection discrétisé :

$$\vec{v}_l \cdot \nabla u \approx v_{l,x} \frac{u_{i,j,l}^n - u_{up_x(i,l),j,l}^n}{\Delta x} + v_{l,y} \frac{u_{i,j,l}^n - u_{i,up_y(j,l),l}^n}{\Delta y}.$$

Le terme intégral de diffusion est discrétisé à l'aide d'une quadrature sur le cercle S^1 avec K directions. On introduit la quantité intégrée :

$$I_{i,j,l}^n(g, u) := \sum_{k=0}^{K-1} w_k u_{i,j,k}^n \Phi_g(v_l, v_k) = \int_{S^1} u^n(t, x, v') \Phi_g(v_l, v') dv'.$$

où w_k sont les poids de quadrature et $\Phi_g(v_l, v_k)$ est la fonction de phase de Henyey–Greenstein pour l'anisotropie g .

Cette discrétisation transforme l'ETR en un système couplé de K équations scalaires, une par direction angulaire v_l . Pour chaque direction $l = 0, \dots, K-1$, on résout l'équation suivante :

$$\frac{u_{i,j,l}^{n+1} - u_{i,j,l}^n}{\Delta t} + v_{l,x} \frac{u_{i,j,l}^n - u_{\text{up}_x(i,l),j,l}^n}{\Delta x} + v_{l,y} \frac{u_{i,j,l}^n - u_{i,\text{up}_y(j,l),l}^n}{\Delta y} + (\mu_a + \mu_s) u_{i,j,l}^n - \mu_s I_{i,j,l}^n = s_{i,j,l}^n.$$

La stabilité numérique du schéma explicite impose la condition CFL :

$$\Delta t \leq \min \left(\frac{\Delta x}{\max_l |v_{l,x}|}, \frac{\Delta y}{\max_l |v_{l,y}|} \right).$$

Cette discrétisation spatiale et temporelle, combinée à la discrétisation angulaire DOM, permet de résoudre efficacement l'équation de transfert radiatif.

3 Filtres de Kalman

3.1 Introduction

Le filtre de Kalman est un algorithme fondamental pour l'estimation récursive de l'état d'un système dynamique à partir de données bruitées. Introduit en 1960 par Rudolf E. Kalman, il fournit une procédure optimale, au sens de la minimisation de l'erreur quadratique moyenne, pour combiner un modèle de prédiction avec des observations incomplètes ou bruitées. Le cadre initial concerne des systèmes linéaires discrets dans le temps, avec des bruits gaussiens blancs sur la dynamique et sur les observations. Dans ce contexte, la mise à jour de l'estimation repose sur des équations fermées et explicites, offrant une solution exacte au problème bayésien linéaire-gaussien.

L'année suivante, Kalman et Bucy généralisent ce formalisme à des systèmes dynamiques continus, donnant naissance au filtre de Kalman-Bucy. Ce dernier a trouvé des applications pionnières dans la navigation inertielle, l'aéronautique ou encore le traitement radar. Rapidement, les limitations du modèle linéaire apparaissent, notamment dans des domaines comme la robotique, la vision par ordinateur, les systèmes biologiques ou les modèles de climat, où les équations d'évolution sont souvent non linéaires. Pour répondre à ces besoins, des variantes du filtre ont été développées, la plus simple étant le filtre de Kalman étendu (EKF), qui repose sur une linéarisation locale du système non linéaire autour de l'état estimé.

D'autres approches ont vu le jour : les filtres ensemblistes (EnKF), fondés sur une représentation stochastique de l'état à l'aide d'un ensemble d'échantillons, et les filtres de particules, qui approchent la loi postérieure de manière non paramétrique. Ces méthodes offrent une meilleure adaptabilité aux dynamiques fortement non linéaires ou non gaussiennes, mais au prix d'un coût numérique accru.

Dans cette étude, nous utilisons le filtre de Kalman étendu, appliqué à un problème inverse régi par une équation de transfert radiatif discrétisée en espace (différences finies) et en angle (DOM). L'état à estimer regroupe le champ lumineux et les paramètres optiques (μ_a, μ_s, g) , tous inaccessibles directement et reconstruits séquentiellement à partir de mesures lumineuses bruitées.

Cette démarche permet d'exploiter la structure dynamique du modèle physique tout en assimilant des données partielles, pour une estimation efficace et robuste des propriétés internes du milieu diffusant.

3.2 Filtre de Kalman linéaire

Dans le cas linéaire, le filtre de Kalman repose sur un modèle dynamique et un modèle d'observation linéaires, de la forme :

$$x_{k+1} = M_{k+1} x_k + w_k, \quad (6)$$

$$y_k = H_k x_k + v_k, \quad (7)$$

où x_k désigne l'état du système à l'instant t_k , y_k la mesure observée, M_{k+1} l'opérateur dynamique, et H_k l'opérateur d'observation. Les bruits w_k et v_k sont supposés gaussiens indépendants, centrés, de matrices de covariance respectives Q_k et R_k .

L'algorithme fonctionne par itérations successives. À chaque pas de temps, une phase de prédiction projette l'état et son incertitude via le modèle dynamique :

$$\mathbf{x}_{k+1}^f = M_{k+1}\mathbf{x}_k^a, \quad \mathbf{P}_{k+1}^f = M_{k+1}\mathbf{P}_k^a M_{k+1}^\top + Q_{k+1}.$$

Puis, lorsqu'une nouvelle observation y_k devient disponible, le filtre la compare à la prédiction via l'innovation $y_k - H_k\mathbf{x}_k^f$, et corrige l'état en la pondérant par un gain \mathbf{K}_k :

$$\mathbf{K}_k = \mathbf{P}_k^f H_k^\top \left(H_k \mathbf{P}_k^f H_k^\top + R_k \right)^{-1},$$

$$\mathbf{x}_k^a = \mathbf{x}_k^f + \mathbf{K}_k \left(y_k - H_k \mathbf{x}_k^f \right),$$

et met à jour la covariance d'erreur :

$$\mathbf{P}_k^a = (I - \mathbf{K}_k H_k) \mathbf{P}_k^f.$$

Ce mécanisme repose sur le fait que, dans le cadre linéaire-gaussien, l'estimateur ainsi obtenu minimise la variance de l'erreur d'estimation. Il correspond à la solution du problème variationnel quadratique suivant :

$$J(x) = \frac{1}{2}(x - x^f)^T P^{-1}(x - x^f) + \frac{1}{2}(y - Hx)^T R^{-1}(y - Hx).$$

En annulant le gradient de J :

$$\nabla J(x) = \frac{\partial J}{\partial x} = P^{-1}(x - x^f) - H^T R^{-1}(y - Hx) = 0,$$

on obtient l'état corrigé x^a et le gain \mathbf{K} :

$$\mathbf{K} = P^f H^T (H P^f H^T + R)^{-1}, \quad \mathbf{x}^a = \mathbf{x}^f + \mathbf{K}(y - H\mathbf{x}^f).$$

Ce gain \mathbf{K} est appelé gain optimal, rend l'estimateur **linéaire sans biais** et **optimal au sens du BLUE** (Best Linear Unbiased Estimator), i.e. il minimise la variance de l'erreur ($\mathbf{x}^a - \mathbf{x}$) parmi tous les estimateurs linéaires sans biais. (i.e. $\mathbb{E}[\mathbf{x}^a - \mathbf{x}] = 0$).

Résumé de l'algorithme du filtre de Kalman

Algorithme du filtre de Kalman

1. Initialisation

- État initial du système : \mathbf{x}_0^f
- Matrice de covariance d'erreur : \mathbf{P}_0^f

2. Pour chaque instant t_k ($k = 1, 2, \dots$) :

(a) Analyse (mise à jour)

1. Calcul du gain de Kalman :

$$\mathbf{K}_k^* = \mathbf{P}_k^f \mathbf{H}_k^\top \left(\mathbf{H}_k \mathbf{P}_k^f \mathbf{H}_k^\top + \mathbf{R}_k \right)^{-1}$$

2. Mise à jour de l'état (analyse) :

$$\mathbf{x}_k^a = \mathbf{x}_k^f + \mathbf{K}_k^* \left(y_k - \mathbf{H}_k \mathbf{x}_k^f \right)$$

3. Mise à jour de la covariance d'erreur :

$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k^* \mathbf{H}_k) \mathbf{P}_k^f$$

(b) Prédiction (forecast)

1. Prédiction de l'état :

$$\mathbf{x}_{k+1}^f = \mathbf{M}_{k+1} \mathbf{x}_k^a$$

2. Prédiction de la covariance d'erreur :

$$\mathbf{P}_{k+1}^f = \mathbf{M}_{k+1} \mathbf{P}_k^a \mathbf{M}_{k+1}^\top + \mathbf{Q}_{k+1}$$

3.3 Extension au cas non linéaire : filtre de Kalman étendu (EKF)

De nombreux systèmes physiques présentent des dynamiques ou des opérateurs d'observation non linéaires. Le filtre de Kalman étendu (EKF) généralise l'algorithme linéaire en linéarisant localement le modèle autour de l'état estimé à chaque itération.

Le modèle devient alors :

$$x_{k+1} = f(x_k) + w_k, \quad y_k = h(x_k) + v_k,$$

où f et h sont des fonctions non linéaires représentant respectivement la dynamique du système et l'opérateur d'observation. À chaque pas k , f et h sont linéarisées au premier ordre autour de l'état courant estimé. On note $F_k = \frac{\partial f}{\partial x} \Big|_{x_k^a}$ le jacobien de la dynamique évalué en x_k^a , et $H_k = \frac{\partial h}{\partial x} \Big|_{x_k^f}$ le jacobien de l'observation évalué en x_k^f .

La phase de prédiction s'écrit :

$$x_{k+1}^f = f(x_k^a), \quad P_{k+1}^f = F_k P_k^a F_k^\top + Q_k,$$

tandis que la mise à jour (analyse) conserve la structure du cas linéaire, mais avec H_k réévalué à chaque pas :

$$K_k = P_k^f H_k^\top (H_k P_k^f H_k^\top + R_k)^{-1}, \quad x_k^a = x_k^f + K_k (y_k - h(x_k^f)), \quad P_k^a = (I - K_k H_k) P_k^f.$$

L'EKF conserve ainsi le schéma récursif du filtre de Kalman et permet d'assimiler séquentiellement des informations dans des modèles non linéaires. Ce cadre méthodologique est exactement celui que nous utiliserons pour la reconstruction conjointe du champ lumineux et des paramètres optiques internes dans la problématique de tomographie optique, par assimilation des mesures issues du modèle de transfert radiatif.

3.4 Interprétation des matrices de covariance et du gain

Dans le cadre de notre approche, trois matrices pilotent la pondération entre la prévision du modèle et l'ajustement aux mesures à chaque étape du filtre : \mathbf{P} , \mathbf{Q} et \mathbf{R} .

Dans notre contexte de tomographie optique, l'état du système est formé à chaque pas de temps par un vecteur $z \in \mathbb{R}^{N+3}$, combinant le champ lumineux $U \in \mathbb{R}^{n \times n \times K}$ (aplatit en un vecteur de taille N) et les trois paramètres optiques μ_a, μ_s, g . Le filtre étendu assimile, à chaque itération, une observation bruitée composée de la moyenne angulaire du champ lumineux en un point (i_0, j_0) du domaine, ainsi que des mesures directes de μ_a, μ_s et g affectées de bruit.

La matrice $\mathbf{P} \in \mathbb{R}^{(N+3) \times (N+3)}$ quantifie l'incertitude sur notre estimation complète de l'état. Sa diagonale contient, pour les N premières composantes, les variances locales sur la distribution spatiale et angulaire de l'intensité lumineuse U , i.e. la confiance que nous avons dans notre estimation actuelle de la propagation des photons. Les trois dernières composantes $\mathbf{P}_{N+i, N+i}$ pour $i = 0, 1, 2$ représentent respectivement l'incertitude sur μ_a, μ_s et g . Si, à l'initialisation, nous considérons ces paramètres comme totalement inconnus, ces valeurs seront fixées élevées (comme 500.0 dans notre cas), tandis que des valeurs faibles reflètent une estimation déjà précise. Les termes extradiagonaux de \mathbf{P} modélisent la corrélation entre l'incertitude sur l'intensité lumineuse et celle des paramètres. Par exemple, une forte corrélation entre U et μ_s signifie que la variabilité de U est fortement influencée par des oscillations de μ_s .

La matrice \mathbf{Q} quantifie l'incertitude associée au modèle de propagation. Ce dernier contient (dans notre cas) un bruit faible pour l'intensité lumineuse U (ordre de 10^{-5}), mais un bruit plus important sur les paramètres μ_a, μ_s, g (ordre de 10^{-3}). Cela reflète l'idée que le schéma DOM utilisé pour simuler U est relativement fiable à chaque pas de temps (avec un petit pas δt), mais que les paramètres optiques peuvent évoluer lentement ou subir de petites dérives non modélisées.

La matrice de bruit d'observation $\mathbf{R} \in \mathbb{R}^{4 \times 4}$ joue un rôle fondamental dans le calibrage du filtre. Dans notre contexte, on a une matrice diagonale, chaque valeur correspondant

à l'erreur attendue sur une composante mesurée. La première valeur (0.01) correspond à l'incertitude sur la mesure de la luminance moyenne au point (i_0, j_0) , tandis que les trois autres reflètent des erreurs sur les observations directes des paramètres. Plus \mathbf{R} est faible, plus le filtre ajuste fortement ses estimations à chaque observation ; si au contraire \mathbf{R} est élevée, on fait plus confiance au modèle de propagation et on ajuste moins l'état aux mesures.

Gain de Kalman.

Dans notre cadre, le gain de Kalman $K \in \mathbb{R}^{(N+3) \times 4}$ est recalculé à chaque itération :

$$K = P^f H^\top (H P^f H^\top + R)^{-1},$$

où P^f est la covariance de l'état prédit, H l'opérateur d'observation et R la covariance du bruit de mesure. Ce gain pondère l'innovation $y - H z^f$ dans la mise à jour :

$$z^a = z^f + K (y - H z^f),$$

et contrôle l'équilibre entre modèle et observation : R faible conduit à une correction fortement influencée par les mesures ; par contre pour un P^f faible donne une correction modérée, le modèle étant jugé fiable.

3.5 Application à la tomographie optique

Dans cette section, on applique le filtre de Kalman étendu (EKF) à un problème inverse. Le but est d'estimer en même temps le champ lumineux U qui évolue dans un milieu diffusant, et les paramètres optiques internes du milieu, qui sont les coefficients d'absorption μ_a , de diffusion μ_s et le paramètre d'anisotropie g . Contrairement à une approche directe où ces paramètres sont supposés connus, l'objectif ici est de les reconstruire progressivement à partir d'un petit nbre de mesures lumineuses bruitées, en intégrant les dynamiques physiques à travers un modèle de transfert radiatif.

On un état augmente du système, noté z_k , qui permet d'inclure à la fois le champ lumineux discrétisé et les paramètres scalaires. L'état augmenté à chaque pas t_k est :

$$z_k = \begin{pmatrix} U_k \\ \mu_a \\ \mu_s \\ g \end{pmatrix} \in \mathbb{R}^{N+3}, \quad N = n \times n \times K$$

où U_k est la luminance discrétisée et vectorisée, et les paramètres sont scalaires.

La composante U_k correspond à la luminance en chque point (x_i, y_j) de la grille et dans chaque direction discrétisée v_k , obtenue selon la méthode DOM. Le tenseur $U_k \in \mathbb{R}^{n \times n \times K}$ est vectorisé en un vecteur colonne de taille N pour le traitement matriciel par le filtre.

L'évolution de U est gouvernée par le modèle direct présenté dans la section 2, discrétisé par un schéma upwind en espace et une quadrature angulaire. À chaque itération, l'état

prédit U_{k+1} est évalué à partir de l'état courant U_k et des paramètres optiques, par une fonction numérique `update_state` implémentant ce schéma. Ce qui constitue la phase de prédiction de l'EKF :

```
1 U_next = update_state(U, mu_a, mu_s, g)
```

Cette fonction intègre simultanément l'advection directionnelle, la diffusion angulaire selon la fonction de Henyey–Greenstein, et les pertes locales dues à l'absorption et à la diffusion hors direction.

Comme le modèle est non linéaire en U mais aussi en (μ_a, μ_s, g) , le filtre de Kalman étendu requiert une linéarisation locale du modèle à chaque itération. Pour ce faire, le jacobien $F_k = \frac{\partial f}{\partial z}$ est calculé automatiquement par différentiation automatique par la bibliothèque PyTorch, comme suit :

```
1 Ju, Jma, Jms, Jg = jacobian(  
2     lambda U,a,s,g: update_state(U, a, s, g),  
3     (u_est, mu_a, mu_s, g),  
4     vectorize=True  
5 )
```

Dns cette implementation, les matrices de dérivées partielles Ju , Jma , Jms , Jg représentent respectivement la sensibilité du champ lumineux à l'état U lui-même, et aux trois paramètres optiques. Ces blocs sont ensuite concaténés pour former la matrice $F_k \in \mathbb{R}^{(N+3) \times (N+3)}$ utilisée pour propager la matrice de covariance d'erreur \mathbf{P} .

L'observation à l'instant t_k est un vecteur $y_k \in \mathbb{R}^4$ contenant la luminance moyenne (sur les directions) mesurée en un point fixe (i_0, j_0) du domaine, ainsi que trois mesures bruitées des paramètres :

$$y_k = \begin{pmatrix} \frac{1}{K} \sum_{p=1}^K U_k(i_0, j_0, p) + \varepsilon_U \\ \mu_a + \varepsilon_{\mu_a} \\ \mu_s + \varepsilon_{\mu_s} \\ g + \varepsilon_g \end{pmatrix}.$$

Cette observation est reliée à l'état z_k via une matrice d'observation linéaire $H \in \mathbb{R}^{4 \times (N+3)}$, dont les coefficients sont construits explicitement : la première ligne combine les K composantes de $U_k(i_0, j_0, \cdot)$ avec un poids $1/K$, tandis que les trois autres lignes extraient directement les paramètres μ_a , μ_s et g .

La matrice \mathbf{R} encode la variance du bruit d'observation. Elle est diagonale, avec une faible variance pour la luminance (i.e. 0.01) et des valeurs plus importantes pour les paramètres optiques (i.e. 10), signifiant qu'on fait plus confiance pour la mesure des paramètres simulés. La matrice \mathbf{Q} quantifie les incertitudes du modèle. Elle est construite avec de faibles valeurs pour les composantes du champ lumineux (i.e. 10^{-5}), reflétant la stabilité du schéma numérique, et des valeurs légèrement plus élevées (i.e. 10^{-3}) pour les trois paramètres, permettant au filtre d'explorer leur évolution à travers les données.

La procédure du filtre EKF suit un enchaînement rigoureux : après la prédiction du champ par le modèle, les jacobiens sont évalués, la covariance \mathbf{P}^f est propagée, puis la mesure est assimilée via le calcul du gain de Kalman \mathbf{K} . L'innovation, c'est-à-dire l'écart

$y_k - Hz_k^f$, est pondérée par ce gain, ce qui permet une mise à jour cohérente de l'état et de sa covariance. Cette étape assure un équilibre optimal entre la dynamique modélisée et les données observées, conformément au principe du BLUE (Best Linear Unbiased Estimator).

Cette méthodologie permet la reconstruction progressive des paramètres internes à partir de données lumineuses partielles, en imposant à chaque étape la cohérence physique du champ lumineux et des incertitudes du modèle.

4 Problème inverse

4.1 Structure du code et approche directe

Pour résoudre le problème inverse en tomographie optique, nous avons développé un simulateur modulaire en **Python**. Il repose sur la discrétisation de l'équation de transfert radiatif (ETR) par la méthode des différences angulaires (DOM).

Ce dernier permet de simuler la propagation de la lumière dans un milieu diffusant, mais aussi une estimation des paramètres optiques (μ_a, μ_s, g) à partir d'observations bruitées par un filtre de Kalman étendu (EKF).

Le code contient plusieurs modules indépendants, chacun est conçu pour une tâche bien précise.

Le fichier `config.py` contient les constantes physiques et numériques globales, telles que la taille de grille $n = 32$, le nombre de directions angulaires $K = 8$, les pas spatiaux $h_x = h_y = 1/(n-1)$, le pas de temps $\Delta t = 10^{-3}$. Ces constantes assurent la reproductibilité et le contrôle du modèle.

Dans `directions.py`, les directions angulaires $\{v_k\}_{k=0}^{K-1}$ sont réparties uniformément sur le cercle unité, avec des poids égaux $w_k = 1/K$. (quadratique trapézoïdale sur $[0, 2\pi]$).

Le fichier `scattering.py` contient le calcul de la fonction de phase de Henyey–Greenstein :

$$\Phi_g(v_l, v_k) = \frac{1 - g^2}{4\pi(1 + g^2 - 2g(v_l \cdot v_k))^{3/2}},$$

normalisée pour garantir la conservation de l'énergie. Cette fonction de phase détermine la manière dont la lumière se diffuse angulairement entre deux directions.

L'évolution du champ lumineux est implémentée dans `model.py`, à travers la fonction `update_state(U_{old}, μ_a, μ_s, g)`. Cette dernière effectue d'abord une advection spatiale selon chaque direction par un schéma upwind, puis applique une décroissance exponentielle due à l'absorption et la diffusion locale, via un facteur multiplicatif $e^{-\Delta t(\mu_a + \mu_s)}$. Enfin, la diffusion angulaire est modélisée par la multiplication tensorielle du champ avec la matrice Φ_g calculée pour la valeur actuelle de g .

Un module auxiliaire `utils.py` complète l'architecture. Il contient notamment l'initialisation de la condition initiale U_0 (sous forme de gaussienne 2D, dupliquée sur K directions), la fonction de saturation `clamp` permettant les valeurs de U , μ_a , μ_s et g d'être dans leurs domaines physiques.

4.2 Application du filtre EKF à la tomographie optique

Le filtre de Kalman étendu est appliqué à un vecteur d'état augmenté :

$$z_k = \begin{pmatrix} U_k \\ \mu_a \\ \mu_s \\ g \end{pmatrix} \in \mathbb{R}^{N+3}, \quad \text{avec } N = n \times n \times K,$$

où U_k est le champ lumineux vectorisé de taille N , et les trois dernières composantes correspondent aux paramètres optiques.

Le cœur de l'algorithme est implémenté dans le fichier `ekf.py`, où la fonction `run_ekf(...)` exécute le filtre récursif. À chaque pas de temps, la dynamique est simulée en appelant `update_state` pour prédire le champ lumineux. Les paramètres (μ_a, μ_s, g) sont considérés constants pendant cette étape. Puis, le jacobien est obtenu par l'autodifférentiation PyTorch :

```

1 Ju, Jma, Jms, Jg = jacobian(
2     lambda U,a,s,g: update_state(U, a, s, g),
3     (u_est, mu_a, mu_s, g),
4     vectorize=True
5 )

```

Le Jacobien complet F est assemblé bloc par bloc, puis utilisé dans la propagation :

$$P^f = F P^a F^T + Q.$$

L'observation utilisée est un vecteur de dimension 4 :

$$y_k = \begin{pmatrix} \frac{1}{K} \sum_{p=1}^K U_k(i_0, j_0, p) + \varepsilon_U \\ \mu_a + \varepsilon_{\mu_a} \\ \mu_s + \varepsilon_{\mu_s} \\ g + \varepsilon_g \end{pmatrix},$$

L'observation $y_k \in \mathbb{R}^4$ contient la luminance moyenne en un point (i_0, j_0) , et les trois paramètres optiques bruités. Elle est reliée à l'état par une matrice H pondérant les K directions par $1/K$ et extrayant directement μ_a, μ_s, g .

Paramétrage du filtre et performances

Pour la configuration initiale, on teste avec des valeurs qui sont très éloignées de la vérité, voici les estimations initiales $(\mu_a, \mu_s, g) = (0.5, 1.0, 0.0)$, alors que les vraies valeurs sont autour de $(0.01, 0.75, 0.8)$. Cette stratégie permet d'évaluer la robustesse du filtre. Les incertitudes initiales sur les paramètres sont fixées à une valeur élevée dans P_0 (par exemple 500), tandis que la variance initiale sur le champ lumineux est modérée.

La matrice Q est diagonale, avec des variances $\mathcal{O}(10^{-5})$ sur U (reflétant la confiance dans le modèle de propagation), et des valeurs plus élevées (par exemple 10^{-3}) pour les paramètres optiques (μ_a, μ_s, g) (évolution plus incertaine).

La matrice de bruit d'observation R est elle aussi configurée de manière différenciée, avec une variance plus faible sur μ_a (ex. 0.001), plus élevée sur μ_s (0.01), et nulle ou quasi nulle sur g , selon les scénarios.

Au fil des itérations, le filtre ajuste les estimations à partir des écarts entre les prédictions et les observations. Le gain de Kalman est recalculé dynamiquement, puis utilisé pour corriger l'état :

$$z^a = z^f + K(y - Hz^f), \quad P^a = (I - KH)P^f.$$

Cette mise à jour permet de réduire l'incertitude sur les paramètres, en tenant compte des mesures disponibles. Les estimations sont projetées dans un domaine admissible par la fonction `clamp`, qui garantit que les valeurs de (μ_a, μ_s, g) restent physiquement réalistes (par exemple, $\mu_a \geq 0$, $\mu_s \geq 0$, $-1 \leq g \leq 1$). Cette étape est cruciale pour éviter des valeurs non physiques qui pourraient perturber la convergence du filtre. Chaque estimation est projetée dans un domaine admissible par la fonction `clamp`, ce qui garantit la stabilité numérique et la cohérence physique.

4.3 Analyse et visualisation des résultats

Le but de cette section est d'illustrer, sur un scénario contrôlé, la capacité du filtre EKF à estimer en même temps les paramètres optiques (μ_a, μ_s, g) à partir de mesures lumineuses bruitées. L'expérience numérique est dans le fichier `main_ekf.py` : on génère un champ de référence U_{true} pour des valeurs physiques réalistes, ensuite on initialise le filtre EKF avec des estimations initiales éloignées de la vérité.

La figure montre l'évolution des estimations de (μ_a, μ_s, g) au cours des itérations du filtre, comparées aux valeurs réelles. Pour faciliter leurs différenciation, une petite perturbation ϵ est parfois ajoutée aux courbes estimées, sans impact sur le calcul ou l'analyse des performances.

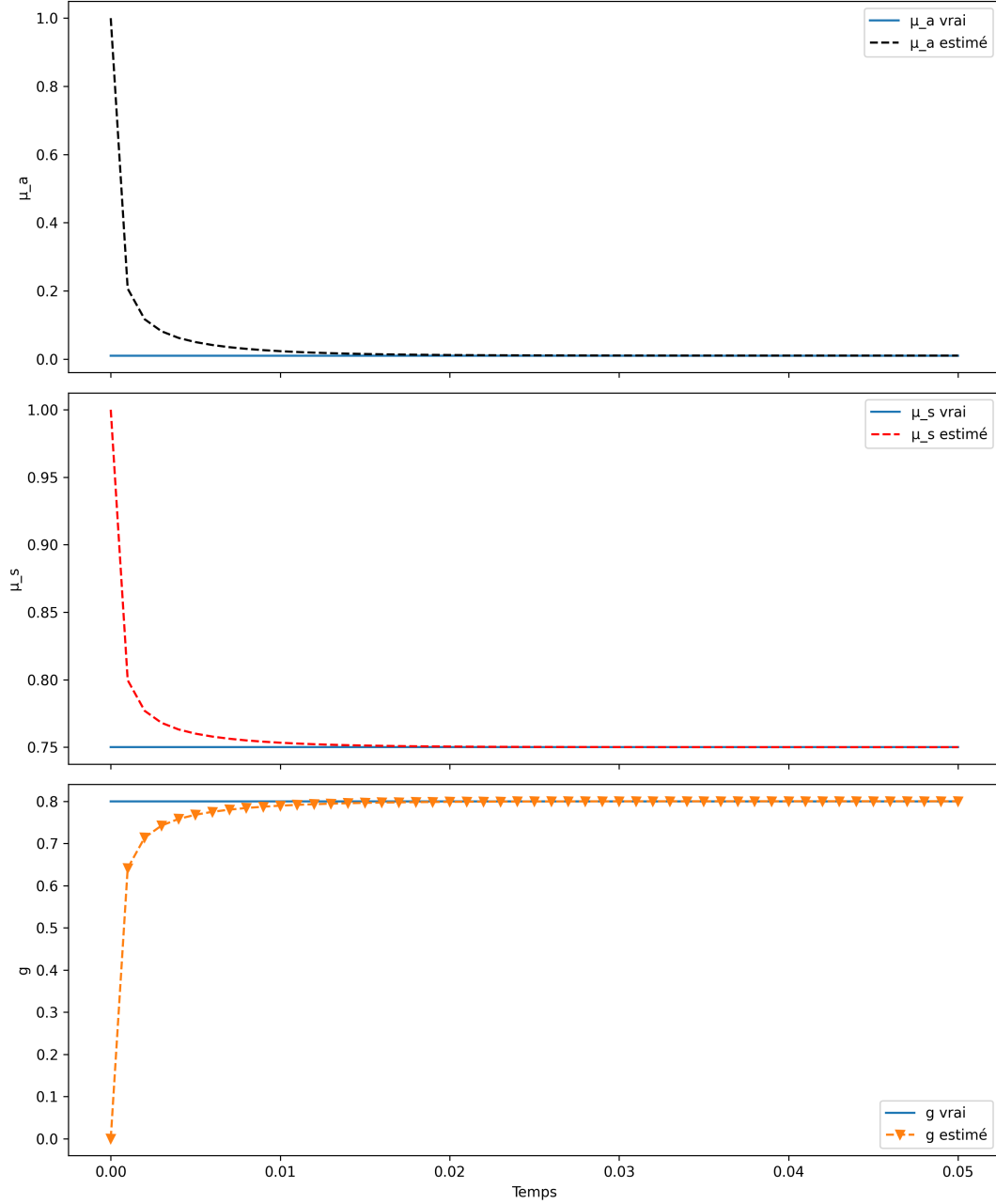


FIGURE 3 – Convergence des paramètres estimés (μ_a, μ_s, g) par le filtre EKF

Cette figure 3 est un exple typique de convergence. On constate que le paramètre μ_a se corrige très rapidement (convergence rapide), tandis que μ_s converge progressivement, et le paramètre d'anisotropie g nécessite plus d'itérations pour atteindre la valeur cible.

L'expérience valide la robustesse du filtre face à une initialisation très éloignée de la vérité. Malgré une initialisation éloignée, la convergence est atteinte en quelques dizaines d'itérations sans instabilité majeure. Les effets du bruit, du choix des matrices Q et R , où des conditions extrêmes, sont analysés dans la suite.

Remarques sur l'implémentation

Dans notre implémentation, on utilise la bibliothèque PyTorch offrant plusieurs avantages.

D'abord, l'autodifférentiation automatique permet un calcul exact des jacobiens nécessaires à l'EKF. Cette dernière garantit la conformité entre le schéma numérique de simulation et sa linéarisation. Ce qui évite tout risque d'erreur ou d'incohérence lors d'une dérivation manuelle du schéma numérique. Cette approche permet assure également une meilleure reproductibilité et facilite l'évolution du code vers des extensions futures.

Les opérations vectorisées offertes par PyTorch améliorent les performances la performance en particulier lors du traitement d'état de grande dimension U de taille $N = n \times n \times K$. Grâce au support GPU, les produits matriciels et les calculs de jacobiens sont accélérés. Ce qui donne la possibilité de traiter des simulations de grande dimension.

Il faut cependant noter que le calcul des jacobiens pour des problèmes de grande taille présente un coût de mémoire et temps non négligeable.

4.4 Calcul de la jacobienne du schéma

Dans le cadre du filtre de Kalman étendu, la linéarisation de la dynamique revient à approximer la propagation du champ

$$U^{n+1} = f(U^n, \mu_a, \mu_s, g)$$

par un modèle localement affine.

La matrice jacobienne $F = \frac{\partial f}{\partial z}$ avec l'état augmenté $z = (U^n, \mu_a, \mu_s, g)$ de dimension $N + 3$, où $N = n \times n \times K$, s'écrit comme suit :

$$F = \begin{pmatrix} \partial_U f & \partial_{\mu_a} f & \partial_{\mu_s} f & \partial_g f \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

où les blocs diagonaux inférieurs expriment que les paramètres optiques sont considérés constants d'une étape à l'autre (modèle stationnaire).

Le calcul analytique de cette jacobienne complète, bien que possible à partir de la forme discrète du résidu du schéma (voir Annexe A), s'avère rapidement complexe à dériver et maintenir. Nous faisons donc le choix d'un calcul numérique exact, en utilisant l'autodifférentiation offerte par PyTorch.

L'implémentation s'appuie sur la fonction `torch.autograd.functional.jacobian`, qui permet de dériver automatiquement la fonction de mise à jour `update_state` selon ses quatre entrées. Le fragment de code suivant montre comment cette opération est réalisée :

```
1 Ju, Jma, Jms, Jg = torch.autograd.functional.jacobian(
2     lambda U_, a_, s_, g_: update_state(U_, a_, s_, g_),
```

```

3     (u_est, mu_a, mu_s, g),
4     vectorize=True
5 )
6
7 F = torch.zeros((N+3, N+3), device=device, dtype=dtype)
8 F[:N, :N] = Ju.reshape(N, N)
9 F[:N, N] = Jma.flatten()
10 F[:N, N+1] = Jms.flatten()
11 F[:N, N+2] = Jg.flatten()
12 F[N:, N:] = torch.eye(3, device=device)

```

Listing 1 – Calcul automatique de la Jacobienne et assemblage de F

Cette approche garantit la cohérence entre la dynamique numérique et la linéarisation, sans nécessiter de calcul analytique, et reste adaptable en cas de modification du schéma sous-jacent. Le coût en calcul est élevé, en particulier pour $\partial_U f$ (matrice 8192×8192 pour $n = 32$, $K = 8$), mais l’option `vectorize=True` permet l’exécution efficace sur GPU.

Le calcul formel des dérivées analytiques, utile pour l’interprétation physique (effet voisins, pertes, diffusion), est détaillé en Annexe A.

Quelques points-clés :

1. Grande taille du Jacobien

Pour $n = 32$ et $K = 8$, l’état U compte $32 \times 32 \times 8 = 8192$ valeurs : son Jacobien $\partial_U f$ est donc une matrice 8192×8192 . C’est le calcul le plus lourd (en temps et en mémoire).

2. Accélération par vectorisation et GPU

L’option `vectorize=True` permet à PyTorch de regrouper les dérivations en gros blocs optimisés, et un GPU peut effectuer ces opérations massivement en parallèle, ce qui réduit fortement le temps de calcul.

Le recours à l’autodifférentiation permet de linéariser tous types de schéma de propagation, et d’adapter facilement le filtre à des variantes plus complexes ou à d’autres paramètres.

4.5 Estimation des paramètres optiques par EKF

Pour estimer simultanément les paramètres optiques (μ_a, μ_s, g) à partir des données simulées, nous utilisons un filtre de Kalman étendu (EKF) appliqué à un état augmenté. Cet état, noté $z_k \in \mathbb{R}^{N+3}$, regroupe le champ discret U_k sous forme vectorielle de taille $N = n \times n \times K$, ainsi que les trois paramètres scalaires $\mu_{a,k}$, $\mu_{s,k}$ et g_k que l’on cherche à estimer.

Modèle de prédiction

L’évolution de l’état repose sur la propagation du champ lumineux par la fonction numérique `update_state`, notée ici f . À chaque itération k , le champ est mis à jour par la relation :

$$U_{k+1|k} = f(U_k, \mu_{a,k}, \mu_{s,k}, g_k),$$

tandis que les paramètres optiques sont supposés constants, c'est-à-dire :

$$\mu_{a,k+1|k} = \mu_{a,k}, \quad \mu_{s,k+1|k} = \mu_{s,k}, \quad g_{k+1|k} = g_k.$$

L'ensemble est alors réuni dans l'état prédit $z_{k+1|k} = [U_{k+1|k}, \mu_{a,k}, \mu_{s,k}, g_k]$.

Linéarisation et prédiction de la covariance

Pour propager l'incertitude à travers la dynamique non linéaire, on linéarise le modèle autour de l'état courant. Le jacobien F_k du schéma de mise à jour est calculé par autodifférentiation automatique vectorisée(PyTorch)

```

1 Ju, Jma, Jms, Jg = jacobian(
2     lambda U_, a_, s_, g_: update_state(U_, a_, s_, g_),
3     (U_k, mu_a_k, mu_s_k, g_k),
4     vectorize=True
5 )

```

Listing 2 – Calcul vectorisé du Jacobien

Cette fonction permet d'obtenir les dérivées partielles de la fonction f par rapport au champ lumineux U_k et aux paramètres optiques (μ_a, μ_s, g) . On assemble la matrice de transition linéarisée ainsi :

$$F_k = \begin{pmatrix} \frac{\partial f}{\partial U_k} & \frac{\partial f}{\partial \mu_a} & \frac{\partial f}{\partial \mu_s} & \frac{\partial f}{\partial g} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

où le premier bloc est de taille $N \times N$.

On fixe ensuite la matrice de covariance initiale $P_0 = \text{diag}(\underbrace{0.01, \dots, 0.01}_N, 500, 500, 500)$,

où les premières N valeurs correspondent à une faible incertitude sur le champ lumineux, tandis que les trois dernières composantes montrent une forte incertitude sur les paramètres optiques. La matrice de bruit de modèle $Q = \text{diag}(\underbrace{1e-5, \dots, 1e-5}_N, 1e-3, 1e-3, 1e-3)$

est choisie pour refléter une confiance modérée sur les paramètres, avec des incertitudes faibles sur le champ lumineux et modérées sur les paramètres.

La propagation de la covariance suit alors la formule classique du filtre de Kalman :

$$P_{k+1|k} = F_k P_k F_k^\top + Q,$$

où P_k est la covariance à l'instant k , et Q est la matrice de bruit de modèle. Ce qui permet au filtre de réajuster ses incertitudes à chaque étape.

Dans notre cas, elle est choisie diagonale, avec des incertitudes faibles sur le champ (10^{-5}) et modérées sur les paramètres (typiquement 10^{-3}). La matrice P_0 initiale est faite avec une faible incertitude sur U (valeurs proches de 0.01), mais on a une forte incertitude sur les paramètres (par exemple diagonale égale à 500), afin de tester la capacité du filtre à corriger une estimation initiale éloignée.

Observation et matrice H

L'observation consiste à mesurer la moyenne angulaire du champ lumineux en un point fixe (i_0, j_0) :

$$\bar{U}_{k+1} = \frac{1}{K} \sum_{l=1}^K U_{k+1|k}(i_0, j_0, l),$$

et à ajouter les valeurs simulées (bruitées) des trois paramètres optiques. Cette observation est représentée sous forme vectorielle $y_{k+1} \in \mathbb{R}^4$, ce qui impose que la matrice d'observation $H \in \mathbb{R}^{4 \times (N+3)}$. Les trois dernières lignes de H sont triviales (extraction directe des paramètres), tandis que la première ligne applique une pondération $1/K$ sur les K composantes de U correspondant à la position spatiale (i_0, j_0) .

Correction de l'état estimé

Avec la mesure y_{k+1} , le filtre corrige son estimation en trois étapes classiques. On commence par calculer l'innovation :

$$e = y_{k+1} - H z_{k+1|k},$$

puis la matrice de covariance de l'observation :

$$S = H P_{k+1|k} H^\top + R,$$

où R est la matrice des variances d'observation. Le gain de Kalman est alors :

$$K_k = P_{k+1|k} H^\top S^{-1}.$$

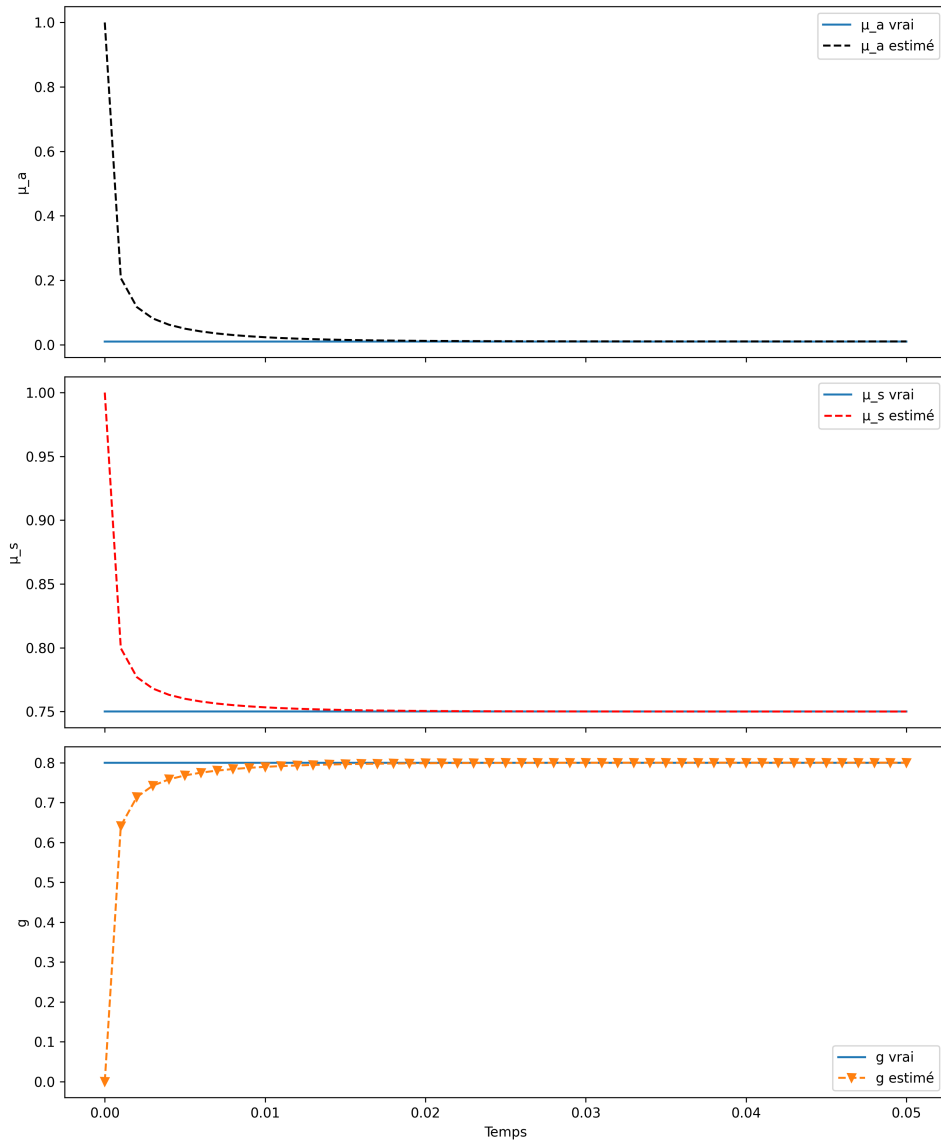
L'état est corrigé par :

$$z_{k+1} = z_{k+1|k} + K_k e, \quad P_{k+1} = (I - K_k H) P_{k+1|k}.$$

On applique ensuite une projection sur les variables : le champ U est contraint à rester positif, et les paramètres doivent respecter les bornes physiques $(\mu_a, \mu_s) \geq 10^{-6}$, $g \in [-1, 1]$.

Boucle EKF et convergence

Les étapes précédentes sont répétées à chaque pas de temps dans une boucle de filtrage récursive. L'évolution dans le temps des estimations $\hat{\mu}_a(t)$, $\hat{\mu}_s(t)$ et $\hat{g}(t)$ montre la capacité du filtre à converger vers les bonnes valeurs, malgré une incertitude initiale forte et la présence de bruit. Une illustration typique est présentée en figure 4.

FIGURE 4 – Convergence des paramètres estimés $\hat{\mu}_a(t)$, $\hat{\mu}_s(t)$, $\hat{g}(t)$

Remarques pratiques

Ce type de filtre permet une exploration fine du compromis entre la confiance accordée au modèle (via Q) et celle accordée aux mesures (via R). Il est possible de modifier les conditions initiales des paramètres pour évaluer la robustesse de la méthode. L'état U étant inclus dans z , il est possible de visualiser son évolution en extrayant les composantes correspondantes. Enfin, le GPU est utilisé pour accélérer les opérations coûteuses comme le calcul du Jacobien ou les produits de matrices dans la mise à jour de la covariance.

5 Résultats numériques

5.1 Présentation générale

Afin d'évaluer numériquement les performances du filtre de Kalman étendu (EKF) dans l'estimation des paramètres optiques (μ_a, μ_s, g) , une série de simulations contrôlées est mis en œuvre. Le script principal `main_ekf.py` gère l'intégration temporelle du schéma de transport par la fonction `update_state`, ainsi que l'ensemble des étapes de l'EKF : prédiction, observation, correction.

L'état du système à l'instant k est représenté sous la forme d'un vecteur augmenté

$$z_k = \begin{pmatrix} U_k \\ \mu_{a,k} \\ \mu_{s,k} \\ g_k \end{pmatrix} \in \mathbb{R}^{N+3},$$

où $N = n \times n \times K$ correspond à la taille du champ lumineux discretisé sur une grille spatiale (n, n) avec K directions angulaires, aplati sous forme vectorielle. La champ lumineux U_k lui-même est un tenseur de dimension (n, n, K) vectorisé pour un traitement matriciel efficace.

À chaque itération, le filtre fournit une estimation $z_k = [U_k, \mu_{a,k}, \mu_{s,k}, g_k]$, qui est stockée pour analyse ultérieure. La procédure itérative de prédiction-correction s'exécute sur un nbre fixé de pas de temps. Le suivi des trajectoires temporelles des trois paramètres $\hat{\mu}_a(k\Delta t)$, $\hat{\mu}_s(k\Delta t)$, et $\hat{g}(k\Delta t)$ permet de tracer et d'évaluer la vitesse de convergence, la stabilité ainsi que la sensibilité aux divers réglages du filtre.

Nous mettrons en lumière le rôle déterminant joué par les matrices de bruit de modèle Q et d'observation R , qui modélisent respectivement la covariance a priori sur l'évolution dynamique et sur les mesures bruitées. Ces matrices contrôlent le compromis entre la confiance dans le modèle numérique et celle dans les données expérimentales. Leur influence est différente selon chacun des paramètres, lesquels présentent des effets variables sur la dynamique du champ lumineux U et sur leur observabilité respective.

Dans la suite de cette section, nous interprétons en détail les dynamiques de convergence selon les mécanismes physiques sous-jacents (absorption, diffusion, anisotropie) et les propriétés numériques du schéma de transport radiatif, en incluant les réglages du filtre (P_0, Q, R) et les conditions initiales des paramètres.

Enfin, nous présentons une série d'expériences visant à tester la robustesse du filtre face à des conditions extrêmes, notamment une mauvaise initialisation, une augmentation du bruit sur les mesures, ou des variations dans les niveaux de confiance accordés au modèle.

5.2 Analyse détaillée des courbes de convergence

Dans cette section, nous analysons le comportement temporel des paramètres estimés par le filtre de Kalman (EKF) appliqué au modèle de transport radiatif en milieu diffusant.

L'état du système contient à la fois le champ lumineux U_k et les trois paramètres scalaires (μ_a, μ_s, g) que l'on souhaite identifier. La dynamique du champ, définie par la fonction `update_state`, contient ces paramètres à l'évolution spatiale et directionnelle du champ à travers un schéma d'advection-diffusion avec terme de diffusion angulaire.

Le filtre EKF est initialisé de manière volontairement éloignée des vraies valeurs, avec notamment

$$(\mu_a, \mu_s, g) = (0.5, 1.0, 0.0),$$

alors que les vraies valeurs sont de l'ordre de $(0.01, 0.75, 0.8)$. Cette stratégie permet de tester la robustesse du filtre à corriger ses erreurs initiales importantes.

À chaque pas de temps, une mise à jour corrige l'état estimé à partir de la mesure (composée de la moyenne du champ au point central et des vraies valeurs bruitées de μ_a, μ_s, g), suivie d'une prédiction pour l'itération suivante.

Convergence du coefficient d'absorption μ_a .

Le paramètre μ_a agit directement sur l'atténuation exponentielle du champ lumineux, ce qui introduit un impact immédiat sur l'amplitude globale de U . En pratique, toute erreur sur μ_a se traduit par une discordance sensible dans la mesure de du champ, ce qui reflète un gain de Kalman élevé associé à ce paramètre. La correction de μ_a est ainsi très rapide, dès les premières itérations du filtre, comme la forte sensibilité des dérivées $\partial f / \partial \mu_a$ le montre.

Dans la simulation, une estimation initiale de $\hat{\mu}_a(0) = 0.99$ (alors que la valeur réelle est $\mu_a = 0.01$) est corrigée très rapidement par le filtre, malgré une grande différence entre les deux valeurs. Mathématiquement, le gain est obtenu à partir de la matrice $K_k = P_{k+1|k} H^\top S^{-1}$.

Ce comportement est contrôlé notamment par la matrice de bruit d'observation

```
1 R = torch.diag(torch.tensor([0.01, 10.0, 10.0, 10.0], device=device))
```

où la première valeur (0.01) correspond à l'observation du champ U , et les trois suivantes aux paramètres optiques. Cette pondération faible sur U permet au filtre de réagir fortement à toute discordance entre mesure et prédiction.

Convergence du coefficient de diffusion μ_s .

Contrairement à μ_a , μ_s influence indirectement U par la redistribution angulaire de l'énergie grâce à la fonction de phase de Henyey–Greenstein. Cette influence plus subtile se traduit par une sensibilité différente, avec un gain de Kalman plus faible pour μ_s . Ce qui entraîne une correction progressive plus lente et susceptible d'oscillations lorsque la confiance accordée à la mesure (variance dans $R_{3,3}$) est trop élevée.

Pour la simulation, la variance élevée sur la mesure de μ_s évite une sur-correction instable :

$$R_{3,3} = 10.0.$$

Cette valeur permet de modérer l'impact des mesures sur la dynamique du filtre, ce qui est crucial pour éviter des oscillations indésirables dans l'estimation de μ_s .

Par ailleurs, on peut améliorer la convergence en augmentant la variance du bruit de modèle due μ_s dans Q par exemple :

$$Q[N+1, N+1] = 1e-3$$

Ce qui rend le filtre plus souple et lui permet d'explorer un espace d'état plus large.

Ce compromis entre rigidité du modèle (via Q) et fiabilité de la mesure (via R) permet de réguler finement la convergence. La figure 5 présente le comportement observé pour $\hat{\mu}_a$ et $\hat{\mu}_s$.

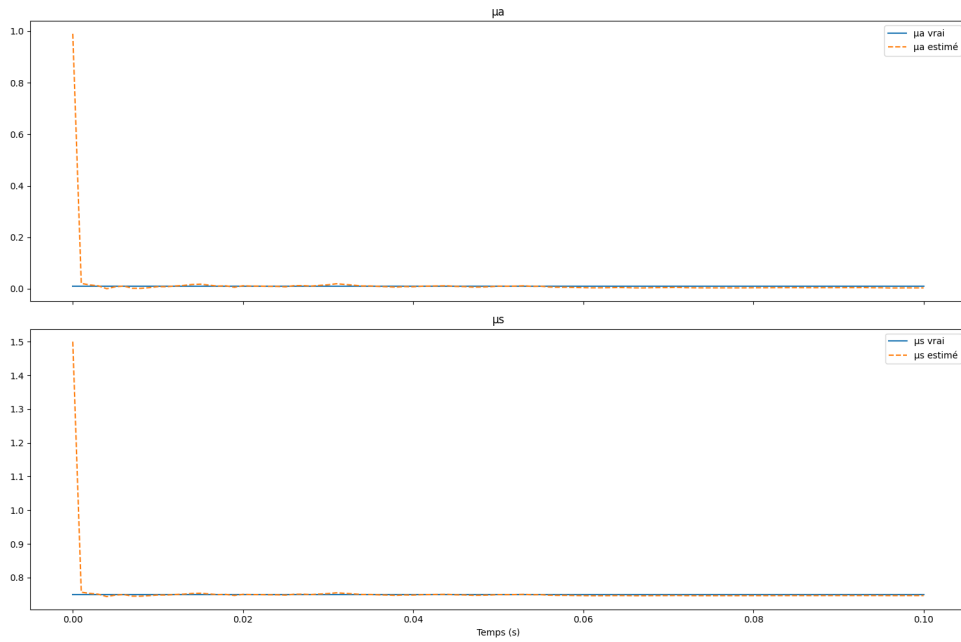


FIGURE 5 – Convergence des estimations $\hat{\mu}_a(t)$ et $\hat{\mu}_s(t)$

Convergence du paramètre de phase angulaire g .

Le paramètre g contrôle l'anisotropie de la diffusion, via le noyau angulaire $\Phi_g(v, v')$ utilisé dans la partie scattering du modèle. Sa valeur influe sur la manière dont l'énergie est redistribuée entre les différentes directions : un g proche de 1 signifie que les photons ont tendance à conserver leur direction initiale (diffusion fortement orientée), tandis qu'un g nul correspond à une diffusion isotrope. Dans notre implémentation, le noyau Φ_g est calculé analytiquement dans la fonction `compute_phi(g)`, et entre dans la dynamique via la sommation `torch.einsum('ijk,kl->ijl', U, phi)`.

En pratique, cette dépendance du champ U à g est à la fois non linéaire et indirecte. Contrairement à μ_a , qui module l'amplitude globale du champ et donc affecte immédiatement la mesure, g agit de manière plus subtile, en modifiant le contenu directionnel de U sans nécessairement en changer la norme. Cette faible observabilité de g dans les mesures conduit le filtre à le corriger lentement, surtout lorsque la variance d'observation associée à g est trop grande.

Dans un premier test, nous avons fixé la matrice de bruit d'observation à

```
1 R = torch.diag(torch.tensor([0.01, 10.0, 10.0, 10.0], device=device,
    dtype=dtype))
```

ce qui reflète une confiance relativement faible dans la composante observée de g . Sous ce réglage, la courbe d'estimation $\hat{g}(t)$ présente des oscillations persistantes autour de la valeur réelle, et la convergence est lente. Le filtre s'appuie alors principalement sur le modèle prédictif, qui, dans le cas de g , n'est que faiblement sensible à ses propres erreurs : le Jacobien de la dynamique par rapport à g présente des valeurs relativement petites.

Comme le montre la figure 6, la trajectoire de $\hat{g}(t)$ reste bruitée même si on observe une tendance à converger vers la valeur réelle $g = 0,8$ après plusieurs itérations. Cependant, la convergence est très lente et le filtre oscille, ce qui illustre la difficulté d'estimer un paramètre non linéaire et faiblement observable comme g dans un contexte de diffusion angulaire.

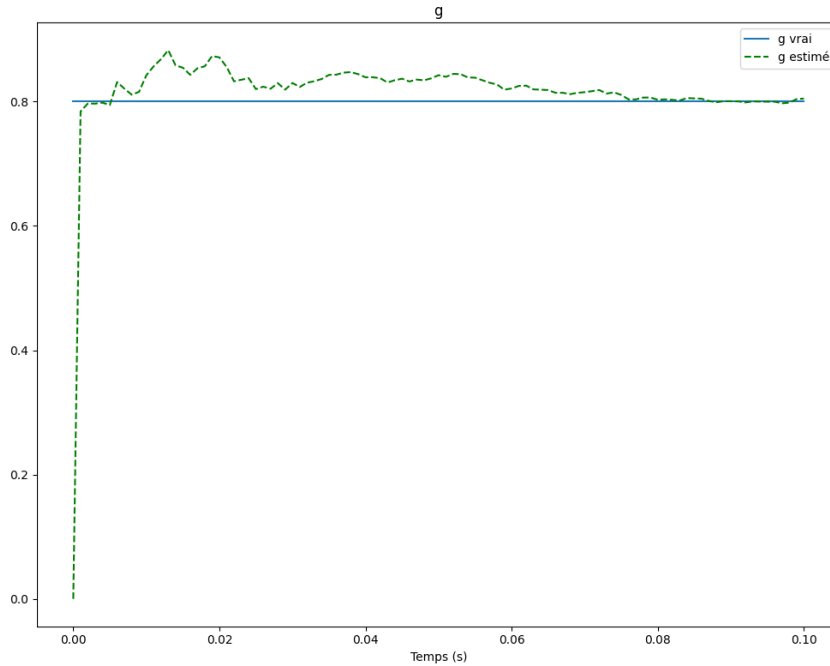


FIGURE 6 – Convergence initiale du paramètre estimé $\hat{g}(t)$ avec $R_{4,4} = 10.0$.

Pour améliorer la performance du filtre sur ce paramètre, nous avons testé un second scénario où l'on augmente la confiance dans la mesure de g , en diminuant sa variance d'observation. On fixe alors :

```
1 R = torch.diag(torch.tensor([0.01, 10.0, 10.0, 1.0], device=device,
    dtype=dtype))
```

Ce changement permet de mieux exploiter la composante mesurée de g à chaque itération du filtre. La figure 7 montre l'effet de cet ajustement : la courbe $\hat{g}(t)$ converge plus rapidement vers la valeur réelle $g = 0,8$, avec une réduction visible du bruit. On constate que la qualité de l'estimation de g dépend donc fortement du paramétrage de

la matrice R , et qu'un bon compromis entre confiance dans la mesure et incertitude du modèle est nécessaire pour obtenir un comportement stable et précis.

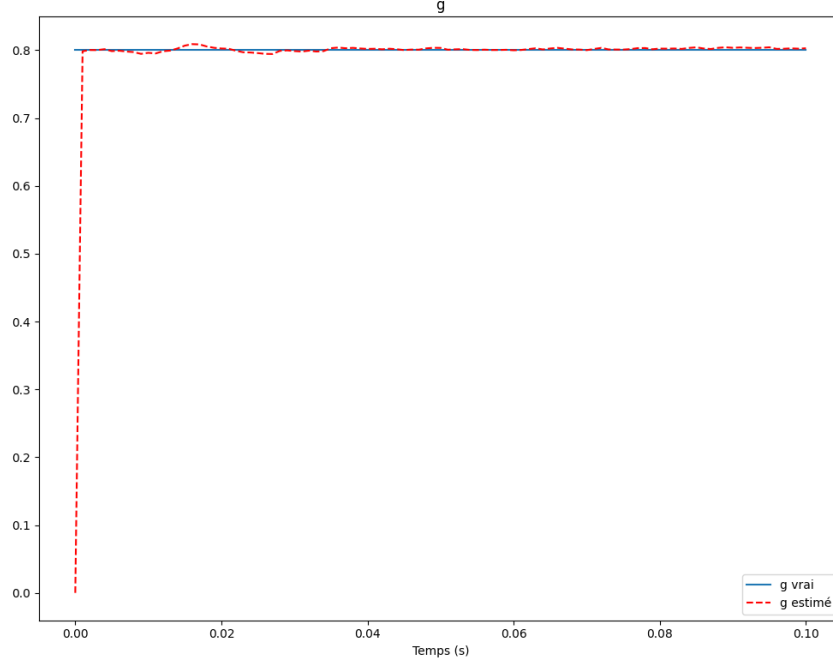


FIGURE 7 – Convergence du paramètre estimé $\hat{g}(t)$ après réduction de $R_{4,4} = 1.0$.

La convergence des paramètres optiques par le filtre de Kalman étendu est fortement influencée par la structure du modèle de transport, la sensibilité des mesures, et les choix de modélisation des bruits Q et R . Chaque paramètre présente des caractéristiques d'observabilité différentes, ce qui se traduit par des dynamiques de convergence variées. L'absorption μ_a est rapidement corrigée, tandis que la diffusion μ_s et l'anisotropie g nécessitent des ajustements plus fins et une confiance adaptée dans les mesures.

Estimation du champ radiatif U :

Au-delà des trois paramètres optiques globaux (μ_a, μ_s, g) , le filtre de Kalman étendu permet de reconstruire dynamiquement l'état complet du champ radiatif U au cours du temps. Ce champ, rappelons-le, est un tenseur de dimension $(n \times n \times K)$. Par vectorisation, on peut reconstituer la structure spatiale et angulaire pour visualiser la distribution lumineuse estimée et la comparer directement au champ de référence U^{true} conforme au modèle physique.

Cette reconstruction joue un rôle de diagnostic, confirmant si les paramètres optique estimés entraînent un comportement cohérent et fidèle au modèle. Des écarts persistants ou des artefacts angulaires dans U peuvent indiquer des erreurs d'estimation ou un mauvais choix des matrices de bruit Q et R .

5.3 Analyse de configurations extrêmes

Afin de mieux cerner le comportement du filtre (EKF) dans des conditions limites, nous avons exploré plusieurs configurations dites « extrêmes », où les matrices de bruit ou l'état initial sont mal ajustés. Ces cas permettent de diagnostiquer finement les sensibilités du filtre, et de proposer des réglages robustes dans un contexte inverse potentiellement mal conditionné.

Une première situation typique est celle d'un filtre doté d'un **gain trop élevé**, c'est-à-dire avec une matrice de bruit d'observation R dont les variances sont très faibles. Cela reflète une forte confiance pour les mesures disponibles. Dans ce cas, le filtre réagit de manière trop agressive à la moindre fluctuation des observations, ce qui se traduit par des oscillations persistantes de l'estimation autour de la valeur vraie, voire par une instabilité numérique si l'écart entre mesure et modèle est trop important.

Cependant, si l'on impose une **confiance excessive dans le modèle**, en fixant le bruit de dynamique Q à des valeurs proches de zéro, le filtre devient rigide. Il prend du temps alors à corriger les erreurs initiales, car les mises à jour sont sous-pondérées par rapport à la propagation du modèle. Ce comportement se manifeste notamment lorsque l'état initial est mal choisi : l'estimation reste bloquée dans une zone de faible vraisemblance, sans rejoindre la vérité même après de nombreuses itérations.

Une troisième configuration critique apparaît lorsque les **bruits de mesure sont supposés trop grands**. Cela revient à fixer des valeurs très élevées dans la diagonale de R , ce qui reflète une faible confiance dans les observations. Dans cette situation, le filtre accorde un poids excessif au modèle prévisionnel, et la convergence vers les paramètres réels devient lente. Ce cas est particulièrement bloquant pour les paramètres faiblement observables (par exple g), qui nécessitent justement un ajustement plus direct sur la mesure pour être corrigés de manière fiable.

Protocole de réglage et bonnes pratiques

La robustesse du filtre repose sur le calibrage précis des matrices d'incertitude initiale (P_0), de dynamique (Q) et d'observation (R). Un protocole méthodique consiste à fixer des ordres de grandeur cohérents avec l'échelle physique des inconnues : P_0 est initialisé avec des variances faibles sur le champ lumineux U (fiabilité du schéma numérique), mais beaucoup plus larges sur les paramètres optiques (μ_a, μ_s, g) , reflétant une grande incertitude initiale sur ces derniers. La matrice Q contient une variance très faible pour U mais plus permissive sur (μ_a, μ_s, g) , afin de permettre leur adaptation progressive au fil de la simulation.

Un point clé est l'usage systématique d'une **projection (clamp)** sur les paramètres estimés : après chaque mise à jour, on impose numériquement que $\mu_a \geq 0$, $\mu_s \geq 0$, $g \in [-1, 1]$. Cette pratique est indispensable pour garantir la cohérence physique et la stabilité même en cas d'initialisation grossière ou de valeurs aberrantes sur les paramètres.

De façon empirique, on ajuste ensuite séparément Q et R en observant les effets sur

les trajectoires. Par exemple, augmenter Q_{gg} rend le filtre plus souple sur le paramètre g , tandis diminuer $R_{4,4}$ permet de mieux exploiter les observations sur g . Varier l'état initial (par exemple, des valeurs très éloignées des vraies valeurs pour (μ_a, μ_s, g)) est essentiel pour tester la capacité du filtre à « revenir » efficacement vers la vérité.

Compléments exploratoires et réglages avancés

Au-delà des scénarios de base analysés dans les sections précédentes, plusieurs expériences numériques complémentaires peuvent être menées pour mieux comprendre les sensibilités du filtre et les effets des différents choix de modélisation. Ces tests permettent notamment de caractériser la manière dont les matrices Q et R influencent la convergence des estimations.

Une première expérience consiste à analyser finement l'effet de la matrice d'observation R , en modifiant individuellement certaines de ses composantes. Par exemple, une réduction de la variance $R_{4,4}$ associée à la mesure du paramètre g conduit à un gain de Kalman plus important pour cette variable. Cela permet d'améliorer généralement la vitesse de convergence. Cependant, une augmentation de $R_{3,3}$, liée à μ_s , permet de tempérer les oscillations qui peuvent apparaître lorsque le filtre suit trop directement les mesures bruitées. Ces ajustements permettent d'équilibrer plus finement la confiance entre modèle et observation selon les paramètres.

Pour le modèle, on peut aussi jouer sur la matrice de bruit Q , notamment sur les éléments diagonaux relatifs aux paramètres optiques. L'élément Q_{gg} , par exple, contrôle la liberté laissée au filtre pour ajuster g entre deux étapes : le multiplier par un facteur $\alpha \in \{0.1, 1, 10\}$ permet d'augmenter ou de restreindre la souplesse du filtre sur ce paramètre. De la même façon, modifier $Q_{N+1,N+1}$, associé à μ_s , permet de contrôler la vitesse de correction de ce paramètre.

5.4 Illustrations et interprétations visuelles

Dans cette partie, les figures qui suivent présentent l'évolution temporelle des paramètres estimés $(\hat{\mu}_a(t), \hat{\mu}_s(t), \hat{g}(t))$ dans différents scénarios, illustrant l'effet de certains réglages dans la configuration du filtre de Kalman étendu (EKF). Cela permet de visualiser la dynamique des estimations ainsi que la robustesse du procédé par rapport aux choix de matrices de bruit ou d'initialisation, conformément à la méthodologie exposée précédemment.

Scénario de référence : convergence avec configuration par défaut.

Dans un premier temps, on considère une configuration de base du filtre où les matrices de bruit sont fixées de manière uniforme, avec une variance de mesure de 10.0 pour chaque paramètre optique :

```
1 R = torch.diag(torch.tensor([0.01, 10.0, 10.0, 10.0], device=device,
    dtype=dtype))
```

et un bruit de modèle homogène de 1×10^{-3} pour les composantes (μ_a, μ_s, g) :

```
1 Q[N:, N:] = torch.eye(3, device=device, dtype=dtype) * 1e-3
```

La figure 8 présente les trajectoires des estimations au fil des itérations. On observe une correction très rapide de μ_a , cohérente avec sa forte influence sur l’amplitude du champ lumineux. Le paramètre μ_s converge plus progressivement, tandis que g demeure relativement bruité. Ce comportement illustre bien la différence d’observabilité des paramètres : μ_a a un impact direct sur la mesure du champ, tandis que μ_s et g ont une influence plus indirecte.

Comme à chaque itération, les valeurs estimées sont projetées dans un domaine physiquement admissible, on s’assure de la cohérence des résultats au fil du filtrage, conformément à la stratégie numérique détaillée précédemment.

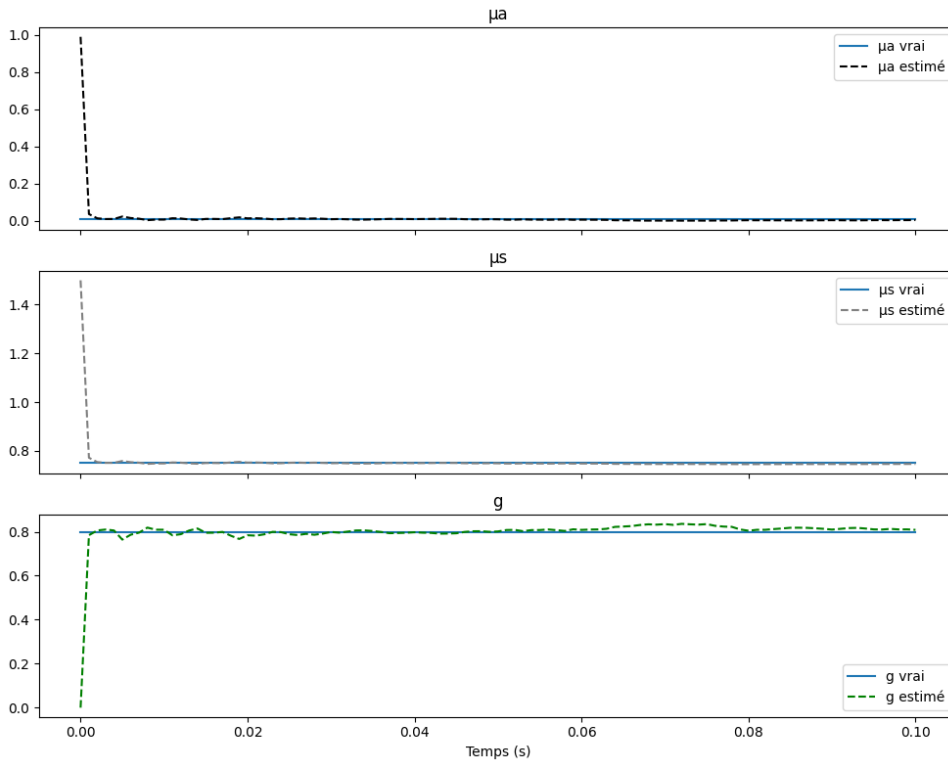


FIGURE 8 – Évolution des estimations $\hat{\mu}_a(t)$, $\hat{\mu}_s(t)$, $\hat{g}(t)$ avec les réglages par défaut.

Effet d’une augmentation du bruit de modèle sur g

Dans une seconde expérience, nous avons exploré l’influence du bruit de modèle associé au paramètre de phase angulaire g . Rappelons que dans la configuration initiale, la composante $Q_{g,g}$ était fixée à une valeur faible (10^{-3}), traduisant une confiance élevée dans la dynamique de g . Toutefois, cette rigidité empêche le filtre de réagir efficacement aux erreurs de prédiction lorsque les mesures contiennent une information utile sur cette variable.

Pour tester l’effet d’un relâchement sur la dynamique de g , nous avons augmenté spécifiquement le terme $Q_{g,g}$, le faisant passer de 10^{-3} à 10^2 , soit un facteur multiplicatif

de 10^5 . Toutes les autres composantes de Q et de R ont été maintenues constantes pour isoler cet effet, conformément aux bonnes pratiques de réglages paramétriques exposées précédemment.

```
1 Q[N+2, N+2] = 100.0 # augmentation de Q_gg
```

La figure 9 illustre clairement les bénéfices de cette modification. Comparée à la situation précédente (cf. figure 8), la courbe estimée $\hat{g}(t)$ devient plus régulière et converge plus rapidement vers la valeur cible $g = 0,8$. On observe une nette réduction des oscillations et un comportement moins bruité, traduisant une meilleure assimilation des observations. L'amélioration observée est d'autant plus significative que le schéma de propagation reste inchangé, ce qui montre le rôle déterminant joué par la matrice Q dans la souplesse du filtre.

Ce test met en évidence une leçon importante : pour des paramètres difficilement observables, par exple g , la convergence ne dépend pas uniquement de la précision des mesures, mais également de la capacité du filtre à s'adapter à l'incertitude du modèle. L'ajustement du bruit de modèle permet ainsi, par une exploration plus large de l'espace des états, de favoriser l'ajustement progressif vers la solution correcte. De plus, comme pour les autres paramètres, chaque mise à jour de l'estimation est suivie d'une projection dans le domaine admissible afin d'assurer la cohérence physique des résultats.

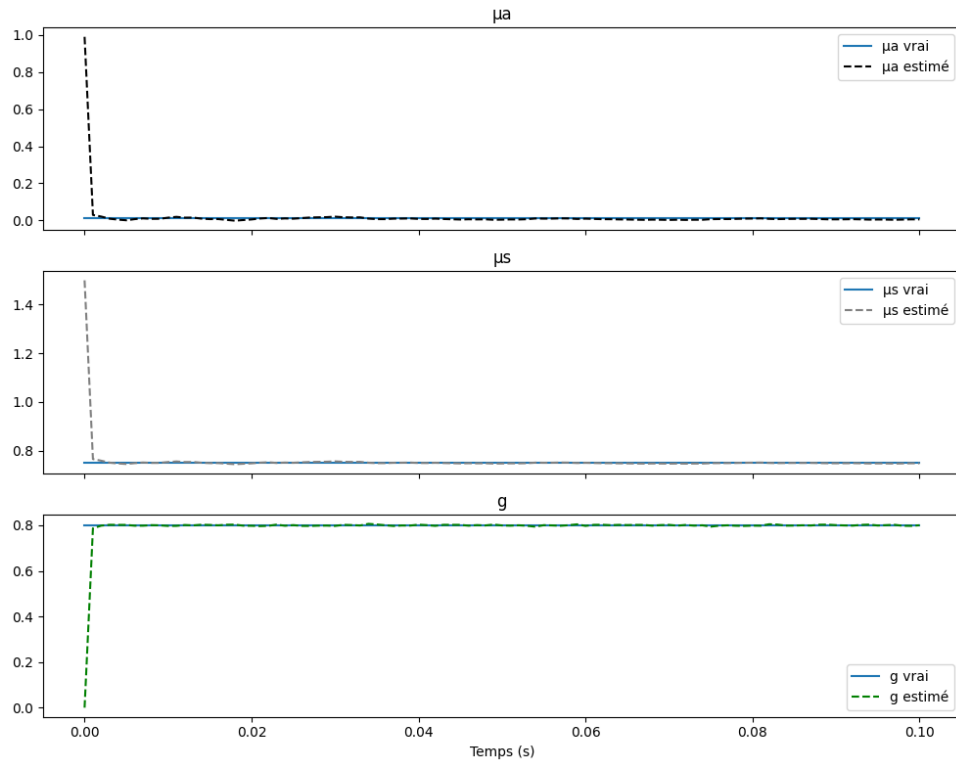


FIGURE 9 – Impact d'une augmentation de $Q_{g,g}$ (de 10^{-3} à 10^2) sur la convergence du paramètre estimé $\hat{g}(t)$.

1) Effet de la covariance d'observation R sur g

Dans la continuité des analyses précédentes, nous avons analysé plus finement la

sensibilité du filtre à la variance d'observation associée au paramètre de phase angulaire g , notée $R_{4,4}$. Comme discuté antérieurement, une valeur trop élevée de cette variance ralentit la convergence de $\hat{g}(t)$. En effet elle limite l'influence des mesures sur la mise à jour de l'état estimé.

Pour étudier ce phénomène de manière systématique, plusieurs simulations ont été réalisées, où seule la valeur de $R_{4,4}$ varie, les autres composantes des matrices R et Q restant constantes. Ce protocole permet d'évaluer isolément l'impact de la confiance accordée à la mesure de g .

Le script Python correspondant procède à une boucle sur différentes valeurs de $R_{4,4}$, chacune étant utilisée pour configurer la matrice R . Pour chaque valeur, une simulation EKF est lancée et l'évolution temporelle des paramètres, notamment g est tracée.

```

1 for R44 in [10.0, 5.0, 1.0, 0.1]:
2     R = torch.diag(torch.tensor([0.01, 10.0, 10.0, R44], device=device,
3         dtype=dtype))
4     zs = run_ekf_simulation(R=R, ... )      # run_ekf_simulation
        plot_parameters(zs, title=f"R44={R44}")

```

Listing 3 – Test : varier $R_{4,4}$

La figure 10 présente les courbes de convergence de $\hat{g}(t)$ pour $R_{4,4} \in \{10.0, 5.0, 1.0, 0.1\}$. On observe que :

1. Pour $R_{4,4} = 10.0$, la courbe présente des oscillations persistantes autour de la valeur cible $g = 0,8$, ce qui traduit une prise en compte limitée des observations dans la correction de l'état.
2. À mesure que $R_{4,4}$ diminue, la courbe devient plus régulière et la convergence vers la valeur vraie s'accélère. notamment, un choix $R_{4,4} = 1.0$ permet une correction est efficace, avec peu de bruit autour de la valeur cible.
3. Lorsque $R_{4,4} = 0.1$, la convergence s'effectue très rapide dès les premières itérations, mais au prix d'une plus grande réactivité aux oscillations de mesure.

Ces résultats confirment que l'estimation du paramètre g dépend fortement du poids attribué à sa mesure. Un calibrage judicieux de $R_{4,4}$ permet ainsi d'optimiser à la fois la vitesse de convergence et sa sensibilité des estimations. Cela illustre de manière remarquable l'importance du compromis entre confiance accordé au modèle (via Q) et celle donnée à l'observation (via R), afin de garantir le bon fonctionnement du filtre dans la cadre d'un problème inverse bruité.

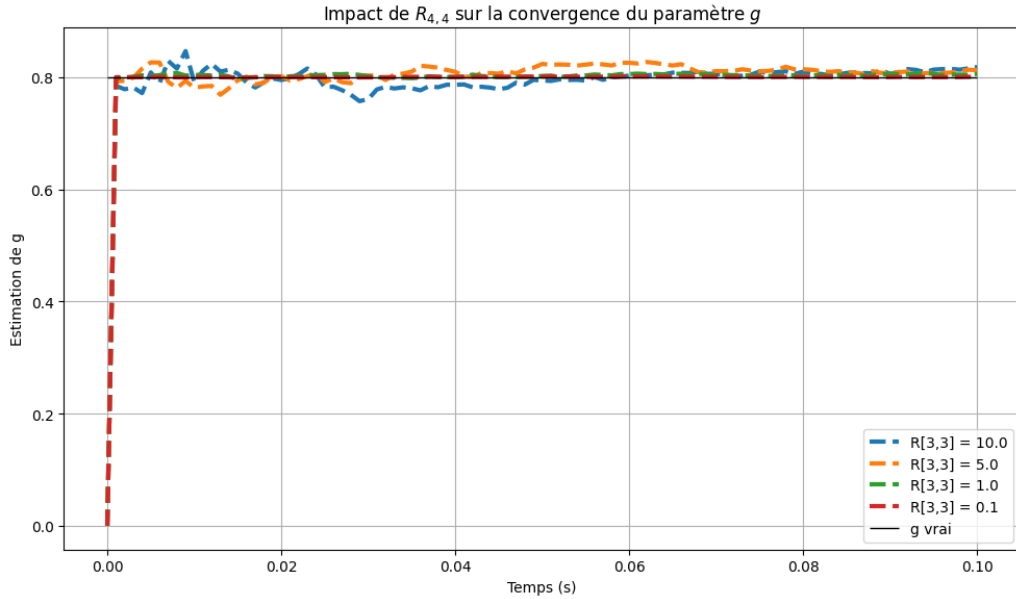


FIGURE 10 – Influence de la variance d’observation $R_{4,4}$ sur la convergence de $\hat{g}(t)$.

2) Impact de la composante du bruit de modèle $Q_{g,g}$

Après avoir analysé l’influence de la matrice de bruit d’observation R , nous nous intéressons ici au rôle du bruit de modèle associée au paramètre de phase angulaire g , c’est-à-dire la composante $Q_{g,g}$ de la matrice Q . Ce test vise à évaluer dans quelle mesure le filtre de Kalman étendu peut améliorer l’ajustement de g en relâchant sa rigidité vis-à-vis de la dynamique du modèle.

Protocole

Trois niveaux croissants de bruit de modèle sur g sont testés, via un facteur multiplicatif $\alpha \in \{0.1, 1.0, 10.0\}$ appliqué à la valeur de base $Q_{g,g} = 10^{-3}$.

Ce bout de code suivant montre comment cette partie est implémentée dans le script Python :

```

1 for alpha in [0.1, 1.0, 10.0]:
2     Q = Q_full.clone()
3     Q[N+2, N+2] = 1e-3 * alpha
4     zs = run_ekf_simulation(Q=Q, ...)
5     plot_parameters(zs, title=f"Qgg factor={alpha}")

```

Listing 4 – Test : variation de $Q_{g,g}$

Interprétation des résultats

Les résultats (voir figure 11) illustrent l’impact direct de du choix de $Q_{g,g}$ sur la convergence et la stabilité de l’estimation du paramètre g :

- Pour $\alpha = 0.1$, la courbe de $\hat{g}(t)$ est très stable, mais la convergence vers la valeur cible est très lente. Le filtre reste peu réactif ; il privilégie la persistance du modèle et corrige faiblement, doù une inertie notable en cas de mauvaise initialisation.

- Pour $\alpha = 1.0$, la convergence s'accélère sensiblement, tout en conservant une trajectoire lisse et peu bruitée. Ce choix représente un bon équilibre entre réactivité et robustesse numérique, limitant le risque de sur-ajustement tout en permettant des corrections efficaces.
- Pour $\alpha = 10.0$, la convergence initiale est très rapide (quasi instantanée), mais la courbe présente ensuite des oscillations légèrement accrues autour de la valeur attendue. Le filtre devient trop réactif à toutes variations des observations ; cela traduit une sensibilité excessive au bruit de mesure, au détriment de la stabilité.

Ce test met en évidence que l'augmentation de $Q_{g,g}$ donne plus de latitude au filtre pour ajuster g , notamment lorsque l'estimation initiale est éloignée de la vérité. Toutefois, Cette liberté accrue doit être modulée : au-delà d'un certain seuil, elle se traduit par des oscillations excessives, qu'il conviendra de compenser par une calibration adaptée de $R_{4,4}$.

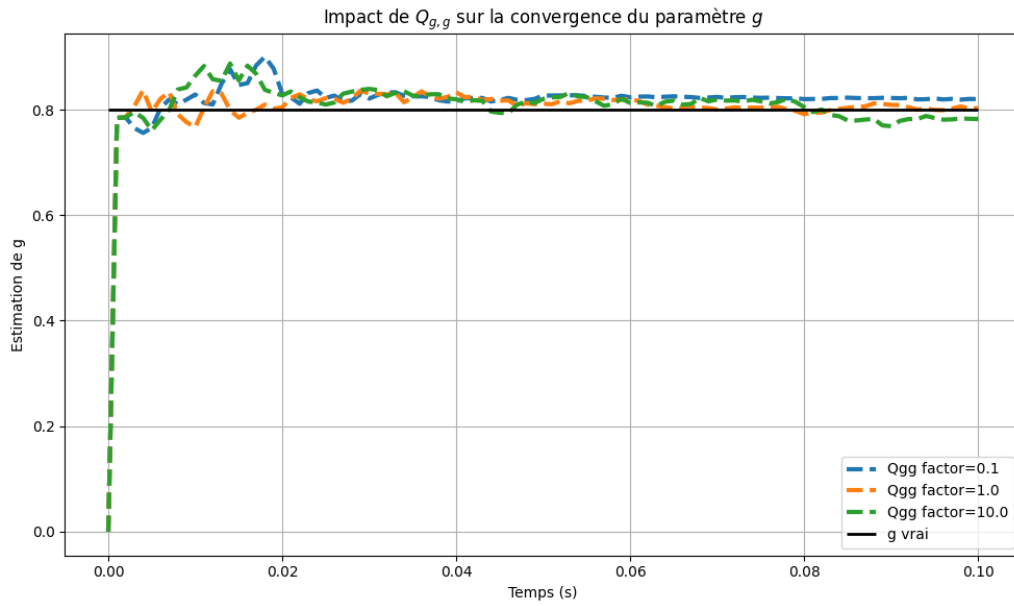


FIGURE 11 – Impact de la composante $Q_{g,g}$ sur la convergence de g

3) Ajout de bruit sur les mesures

Contexte. Dans un contexte expérimental réaliste, les observations issues des capteurs sont entachées de bruit. Pour tester la robustesse du filtre de Kalman étendu face à ces perturbations, nous introduisons explicitement du bruit additif sur la mesure du champ radiatif moyen y_{sum} , ainsi que sur les pseudo-observations des paramètres optiques.

Plus précisément, un bruit gaussien $\varepsilon \sim \mathcal{N}(0, 0.01^2)$ est ajouté à y_{sum} , tandis que les paramètres μ_a et μ_s sont perturbés par un vecteur aléatoire $\varepsilon_i \sim \mathcal{N}(0, \sigma_i^2)$ avec $\sigma = (0.001, 0.01, 0)$, ce qui laisse le paramètre g inchangé.

Dans la boucle de simulation, ces perturbations sont injectées directement dans le vecteur d'observation y_h via le code suivant :

```
1 noise_U      = torch.randn(1, device=device) * 0.01
2 noise_params = torch.tensor([0.001, 0.01, 0.0],
```

```

3                                     device=device) * torch.randn(3, device=
   device)
4 y_h = torch.cat([
5     y_sum + noise_U,
6     torch.tensor([mu_a_true, mu_s_true, g_true], device=device) + noise_
   params
7 ])

```

Listing 5 – Test 4 – Ajout de bruit sur les mesures

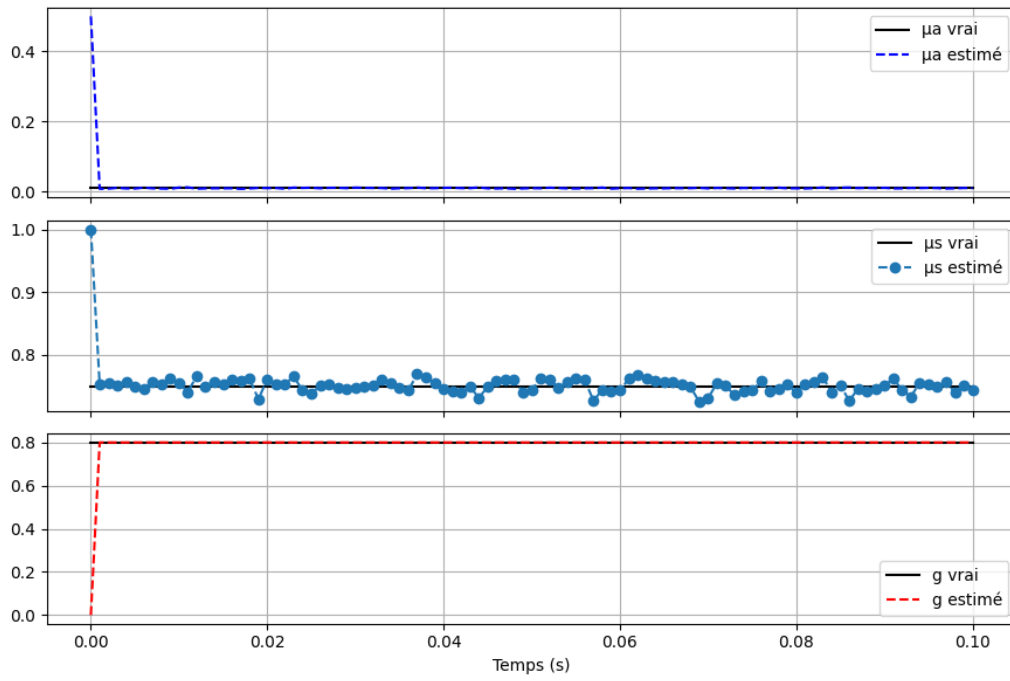
Pour que le filtre tienne compte de cette incertitude supplémentaire, la matrice d'observation R est modifiée de manière cohérente avec les variances injectées :

$$R = \begin{pmatrix} 0.01^2 & 0 & 0 & 0 \\ 0 & 0.001^2 & 0 & 0 \\ 0 & 0 & 0.01^2 & 0 \\ 0 & 0 & 0 & 0^2 \end{pmatrix}$$

Analyse de la figure 12. L'analyse des courbes en présence de bruit sur les observations montre que le paramètre d'absorption μ_a converge rapidement malgré les perturbations, preuve de sa forte observabilité – toute erreur sur ce paramètre se répercute immédiatement sur l'énergie globale du champ. À l'inverse, la diffusion μ_s affiche des oscillations avec une instabilité marquée autour de la valeur réelle. Cette variabilité reflète à la fois l'effet indirect de μ_s sur la mesure principale et le choix d'une variance de bruit relativement élevée sur ce paramètre. Un réglage plus précis de $R_{3,3}$ permettrait de mieux lisser cette trajectoire.

Dans cette configuration où la variance du bruit d'observation associée au paramètre d'anisotropie g est nulle ($R_{4,4} = 0$), l'estimation de g présente une excellente convergence vers la valeur cible, démontrant une forte stabilité et une faible influence des perturbations sur cette variable. Cette stabilité illustre la capacité du filtre à découpler statistiquement chaque variable lorsque la matrice R est bien structurée, évitant ainsi la propagation de bruit entre paramètres disparates.

Enfin, cette expérience confirme l'importance du calibrage des bruits d'observation dans la performance globale de la reconstruction bayésienne à l'aide d'un EKF.

FIGURE 12 – Convergence des estimations $\hat{\mu}_a$, $\hat{\mu}_s$, \hat{g} .

Pour cette figure, la matrice de bruit d'observation est fixée à

$$R = \begin{pmatrix} 0.01^2 & 0 & 0 & 0 \\ 0 & 0.001^2 & 0 & 0 \\ 0 & 0 & 0.01^2 & 0 \\ 0 & 0 & 0 & 0^2 \end{pmatrix}$$

6 Conclusion and Perspectives

Ce travail a porté sur la modélisation et la résolution d'un problème inverse en tomographie optique fondé sur l'équation de transfert radiatif discrétisée. En combinant un schéma numérique explicite upwind dans l'espace et la méthode des ordonnées discrètes pour la discrétisation angulaire, nous avons élaboré un modèle solide de propagation lumineuse dans un milieu diffusant.

L'application d'un filtre de Kalman étendu (EKF), associé à un état augmenté intégrant à la fois le champ lumineux et les paramètres optiques internes (μ_a, μ_s, g) , a permis de reconstruire ces derniers de manière robuste et séquentielle à partir de mesures partielles bruitées. L'utilisation de l'autodifférentiation automatique a facilité le calcul des jacobiens nécessaires à la linéarisation locale, garantissant la cohérence entre modèle et estimation.

Les simulations numériques ont démontré la capacité du filtre à corriger efficacement des estimations initiales éloignées, avec une convergence rapide du coefficient d'absorption, une évolution progressive de la diffusion, et une identification délicate mais améliorée du paramètre d'anisotropie. L'étude de la sensibilité aux choix des matrices de covariance modélisant l'incertitude a mis en évidence le rôle crucial du calibrage entre confiance accordée au modèle et aux observations pour assurer stabilité et précision.

Ces résultats confirment la pertinence de l'EKF comme outil d'estimation bayésienne dans ce cadre non linéaire. Ils ouvrent la voie à des extensions vers des filtres ensemblistes, une meilleure prise en compte des effets non gaussiens, et à l'application sur données expérimentales. Enfin, la modularité de l'implémentation et les principes méthodologiques posés constituent un socle solide pour poursuivre les développements dans le domaine de la tomographie optique inverse.

7 References

Références

- [1] L. G. Henyey and J. L. Greenstein. Diffuse radiation in the galaxy. *Astrophysical Journal*, 93 :70–83, 1941.
- [2] David Le Hardy. *Traitement des conditions aux limites spéculaires pour l'étude du transfert radiatif dans des matériaux à géométrie complexe*. PhD thesis, Nantes, 2017.
- [3] Anne Pallarès, Philippe Schmitt, and Wilfried Uhring. Comparison of time resolved optical turbidity measurements for water monitoring to standard real-time techniques. *Sensors*, 21(9) :3136, 2021.
- [4] Yukio Yamada, Hiroaki Suzuki, and Yutaka Yamashita. Time-domain near-infrared spectroscopy and imaging : A review. *Applied Sciences*, 9(6), 2019.
- [5] Hiroyuki Fujii, Yukio Yamada, Go Chiba, Yoko Hoshi, Kazumichi Kobayashi, and Masao Watanabe. Accurate and efficient computation of the 3d radiative transfer equation in highly forward-peaked scattering media using a renormalization approach. *Journal of Computational Physics*, 374 :591–604, 2018.

- [6] André Liemert and Alwin Kienle. Infinite space green's function of the time-dependent radiative transfer equation. *Biomedical optics express*, 3(3) :543–551, 2012.
- [7] Euractiv. L'oms alerte sur les dangers de l'eau contaminée, 2023.
- [8] OMLC. Definition of μ_a and μ_s , 2024.
- [9] OMLC. Definition of μ_a and μ_s , 2024.
- [10] Winsen Sensor. Turbidity, 2023.
- [11] Maths-Vives. Site officiel de maths-vives.

Annexe : Interprétation et calcul des incertitudes

Pourquoi quantifier l'incertitude ?

Dans tout problème inverse basé sur des données bruitées, il est essentiel de ne pas se contenter d'une estimation ponctuelle (valeur moyenne), mais de fournir également une mesure de l'« incertitude » associée. Cette incertitude reflète le degré de confiance que l'on peut avoir dans l'estimation : une faible incertitude traduit une précision accrue, tandis qu'une incertitude élevée signale une estimation plus fragile.

Rôle de la covariance dans le filtre de Kalman

Le filtre de Kalman met à jour, à chaque pas de temps, une matrice de covariance d'erreur d'estimation notée \mathbf{P}_k , qui joue un rôle central dans la quantification de l'incertitude.

Définition. Pour un vecteur d'état $x_k \in \mathbb{R}^n$, la matrice \mathbf{P}_k est définie par :

$$\mathbf{P}_k = \mathbb{E}[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T],$$

où \hat{x}_k est l'estimation de l'état au pas k .

Interprétation des composantes.

- \mathbf{P}_{ii} : variance de l'estimation de x_i . Une valeur élevée indique une incertitude importante.
- $\sqrt{\mathbf{P}_{ii}}$: écart-type, souvent utilisé pour représenter graphiquement des bandes d'erreur autour de la moyenne estimée.
- \mathbf{P}_{ij} : covariance entre les variables x_i et x_j , indiquant leur interdépendance.

Lien avec les intervalles de confiance

Dans le cas d'un modèle linéaire gaussien, chaque composante x_i de l'état est modélisée comme une variable aléatoire gaussienne :

$$x_i \sim \mathcal{N}(\hat{x}_i, \mathbf{P}_{ii}).$$

Ainsi, un intervalle de confiance à 95% est donné par :

$$\hat{x}_i \pm 1.96 \cdot \sqrt{\mathbf{P}_{ii}}.$$

Autrement dit, si le modèle est statistiquement cohérent, la probabilité que la vraie valeur de x_i se trouve dans cet intervalle est de 95%.

Exemple d'interprétation pratique

Supposons que le filtre retourne l'estimation suivante pour μ_a :

$$\hat{\mu}_a = 0.015, \quad \text{avec } \mathbf{P}_{\mu_a, \mu_a} = 10^{-6}.$$

L'écart-type est alors :

$$\sigma_{\mu_a} = \sqrt{10^{-6}} = 0.001,$$

et l'intervalle de confiance à 95% devient :

$$\mu_a \in [0.015 \pm 1.96 \times 0.001] = [0.01304, 0.01696].$$

Ce type d'information est exploité visuellement en traçant des courbes $t \mapsto \hat{\mu}_a(t)$ accompagnées de bandes d'erreur $\pm 2\sigma(t)$.

Corrélations croisées entre paramètres

Lorsque plusieurs variables sont estimées simultanément, des corrélations peuvent apparaître. Par exemple, si $\mathbf{P}_{\mu_a, \mu_s} \gg 0$, cela signifie qu'une incertitude sur μ_a affecte aussi l'estimation de μ_s , ce qui traduit une interdépendance dans les observations.

Conclusion

La matrice \mathbf{P} constitue une brique essentielle du filtre de Kalman : elle permet non seulement d'exprimer quantitativement la confiance dans les états estimés, mais aussi de contrôler dynamiquement l'actualisation de ces états en fonction du bruit modélisé. Elle fournit ainsi un outil précieux pour diagnostiquer la fiabilité des résultats obtenus dans le cadre d'un problème inverse.

Annexe : Organisation du code EKF

L'implémentation du filtre de Kalman étendu (EKF) a été conçue de manière modulaire afin de faciliter sa maintenance, sa lisibilité et sa réutilisation dans divers scénarios. Le code est structuré en plusieurs fichiers, chacun jouant un rôle spécifique :

1. **config.py** : contient les constantes physiques et les paramètres de simulation (taille de la grille, pas de temps, durée, etc.). Ce fichier centralise les réglages pour rendre les tests reproductibles.
2. **directions.py** : gère la discrétisation angulaire du domaine. Il fournit les directions \vec{v}_l et les poids quadrature w_l , utilisés dans le calcul du champ radiatif et du terme de diffusion.
3. **scattering.py** : définit la fonction de phase Henyey–Greenstein Φ_g , qui modélise l'anisotropie de la diffusion. Ce module contient aussi des outils de visualisation et d'évaluation de la phase.

4. `model.py` : contient la mise à jour explicite du champ radiatif U selon le schéma de transport discrétisé. Il inclut également les fonctions associées à la simulation directe.
5. `utils.py` : regroupe les fonctions auxiliaires telles que :
 - `clamp_parameters` pour forcer $\mu_a, \mu_s > 0$ et $g \in [-1, 1]$,
 - initialisation de l'état et des matrices P_0, Q, R ,
 - interpolation ou sauvegarde des résultats.
6. `ekf.py` : contient l'implémentation complète du filtre de Kalman étendu. Ce module assure la prédiction, la mise à jour et le calcul automatique des jacobiens via PyTorch.
7. `main_ekf.py` : point d'entrée principal du programme. Il orchestre la simulation inverse, exécute les scénarios de test, génère les figures, et sauvegarde les résultats.

Implémentation technique. L'ensemble du projet est implémenté en **PyTorch**, ce qui permet :

- de bénéficier de l'auto-différentiation pour le calcul exact des jacobiens F_k et H_k ,
- d'utiliser les accélérations GPU (via CUDA) pour les simulations intensives,
- de garantir une compatibilité avec des outils modernes de deep learning ou d'optimisation.

Cette organisation modulaire rend le code facilement extensible (par exemple pour des filtres plus avancés comme UKF ou EnKF) et facilite l'exploration numérique dans un cadre pédagogique ou de recherche.

A Calcul formel de la jacobienne du schéma

Dans cette annexe, nous dérivons analytiquement les différentes composantes du jacobien du schéma de mise à jour $f(U^n, \mu_a, \mu_s, g)$. Ces expressions permettent de mieux comprendre l'influence des différents termes physiques sur la sensibilité de la solution, bien que ce jacobien ne soit pas utilisé dans l'implémentation numérique (cf. Section 4.4).

Dérivation formelle du résidu

Pour mieux comprendre la structure de l'upwind, des pertes et du scattering, on considère le résidu du schéma discret

$$\begin{aligned}
 R_{i,j,l} := & \frac{u_{i,j,l}^{n+1} - u_{i,j,l}^n}{\delta t} + \frac{v_{l,x}}{\Delta x} (u_{i,j,l}^n - u_{up(i),j,l}^n) + \frac{v_{l,y}}{\Delta y} (u_{i,j,l}^n - u_{i,up(j),l}^n) \\
 & + (\mu_a + \mu_s) u_{i,j,l}^n - \mu_s \sum_{k=0}^{K-1} w_k u_{i,j,k}^n \Phi_g(v_l, v_k) - s(t^n, x_{i,j}, v_l). \quad (8)
 \end{aligned}$$

Concernant le calcul de la jacobienne du schéma, nous allons dériver l'équation (8) par rapport à $u_{i,j,l}^n, \mu_a, \mu_s$ et g .

1. Dérivée par rapport à $u_{i,j,l}^n$

Les termes dépendant de $u_{i,j,l}^n$ apparaissent dans plusieurs parties dans l'équation (8). Nous allons les dériver un par un.

(a) Terme temporel

Le terme temporel est donné par :

$$\frac{u_{i,j,l}^{n+1} - u_{i,j,l}^n}{\delta t} = \frac{1}{\delta t} (u_{i,j,l}^{n+1} - u_{i,j,l}^n)$$

Lorsque l'on dérive par rapport à $u_{i,j,l}^n$, on obtient :

$$\frac{\partial}{\partial u_{i,j,l}^n} \left(\frac{u_{i,j,l}^{n+1} - u_{i,j,l}^n}{\delta t} \right) = \frac{1}{\delta t} (0 - 1) = -\frac{1}{\delta t}$$

(b) Terme d'advection Le terme d'advection est donné par :

$$\frac{v_{l,x}}{\Delta x} (u_{i,j,l}^n - u_{up(i),j,l}^n) + \frac{v_{l,y}}{\Delta y} (u_{i,j,l}^n - u_{i,up(j),l}^n)$$

i. Pour le terme central $u_{i,j,l}^n$, on a :

$$\frac{\partial}{\partial u_{i,j,l}^n} \left(\frac{v_{l,x}}{\Delta x} (u_{i,j,l}^n - u_{up(i),j,l}^n) + \frac{v_{l,y}}{\Delta y} (u_{i,j,l}^n - u_{i,up(j),l}^n) \right) = \frac{v_{l,x}}{\Delta x} + \frac{v_{l,y}}{\Delta y}$$

ii. Pour les termes voisins « amont » :

$$\frac{\partial R_{i,j,l}}{\partial u_{up(i),j,l}^n} = -\frac{v_{l,x}}{\Delta x}$$

et

$$\frac{\partial R_{i,j,l}}{\partial u_{i,up(j),l}^n} = -\frac{v_{l,y}}{\Delta y}$$

(c) Terme de diffusion angulaire

Le terme de diffusion angulaire est donné par :

$$-\mu_s I_{i,j,l}^n(g, u) = -\mu_s \sum_{k=0}^{K-1} w_k u_{i,j,k}^n \Phi_g(v_l, v_k)$$

Pour une direction donnée k , la dérivée par rapport à $u_{i,j,l}^n$ est donnée par :

$$\frac{\partial}{\partial u_{i,j,l}^n} (-\mu_s I_{i,j,l}^n(g, u)) = -\mu_s w_k \Phi_g(v_l, v_k)$$

Pour k tel que $k = l$, on a :

$$\frac{\partial}{\partial u_{i,j,l}^n} (-\mu_s I_{i,j,l}^n(g, u)) = -\mu_s w_l \Phi_g(v_l, v_l)$$

Et on peut simplifier $\Phi_g(v_l, v_l)$ par :

$$\begin{aligned}\Phi_g(v_l, v_l) &= \frac{1 - g^2}{2\pi(1 + g^2 - 2g)} \\ &= \frac{1 - g^2}{2\pi(1 - g)^2} \\ &= \frac{(1 - g)(1 + g)}{2\pi(1 - g)^2} \\ &= \frac{1 + g}{2\pi(1 - g)}\end{aligned}$$

Ainsi, on a :

$$\frac{\partial}{\partial u_{i,j,l}^n} (-\mu_s I_{i,j,l}^n(g, u)) = -\mu_s w_l \frac{1 + g}{2\pi(1 - g)}$$

- (d) **Terme de réaction local(absorption et diffusion locale)** Ce terme est donné par :

$$(\mu_a + \mu_s)u_{i,j,l}^n$$

D'où la dérivée par rapport à $u_{i,j,l}^n$ est donnée par :

$$\frac{\partial}{\partial u_{i,j,l}^n} ((\mu_a + \mu_s)u_{i,j,l}^n) = (\mu_a + \mu_s)$$

Donc la dérivée totale par rapport à $u_{i,j,l}^n$ est donnée par :

$$\frac{\partial R_{i,j,l}}{\partial u_{i,j,l}^n} = -\frac{1}{\delta t} + \frac{v_{l,x}}{\Delta x} + \frac{v_{l,y}}{\Delta y} + (\mu_a + \mu_s) - \mu_s w_l \Phi_g(v_l, v_l)$$

Pour les termes voisins amont, on a :

$$\begin{aligned}\frac{\partial R_{i,j,l}}{\partial u_{up(i),j,l}^n} &= -\frac{v_{l,x}}{\Delta x} \\ \frac{\partial R_{i,j,l}}{\partial u_{i,up(j),l}^n} &= -\frac{v_{l,y}}{\Delta y}\end{aligned}$$

Pour les autres directions $k \neq l$, on a :

$$\frac{\partial R_{i,j,l}}{\partial u_{i,j,k}^n} = -\mu_s w_k \Phi_g(v_l, v_k)$$

2. Dérivée par rapport aux coefficients μ_a, μ_s

(a) **Pour μ_a**

Le coefficient μ_a apparaît uniquement dans le terme de réaction local :

$$(\mu_a + \mu_s)u_{i,j,l}^n$$

D'où :

$$\frac{\partial R_{i,j,l}}{\partial \mu_a} = u_{i,j,l}^n$$

(b) **Pour** μ_s

Le coefficient μ_s apparaît dans deux contributions :

i. Dans le terme local :

$$(\mu_a + \mu_s)u_{i,j,l}^n$$

ce qui implique :

$$\frac{\partial}{\partial \mu_s} ((\mu_a + \mu_s)u_{i,j,l}^n) = u_{i,j,l}^n$$

ii. Dans le terme de diffusion angulaire :

$$-\mu_s I_{i,j,l}^n(g, u) = -\mu_s \sum_{k=0}^{K-1} w_k u_{i,j,k}^n \Phi_g(v_l, v_k)$$

D'où :

$$\frac{\partial}{\partial \mu_s} (-\mu_s I_{i,j,l}^n(g, u)) = -I_{i,j,l}^n(g, u)$$

Donc la dérivée totale par rapport à μ_s est donnée par :

$$\frac{\partial R_{i,j,l}}{\partial \mu_s} = u_{i,j,l}^n - I_{i,j,l}^n(g, u)$$

3. Dérivée par rapport au paramètre de phase g

Le paramètre de phase g apparaît uniquement dans la fonction de phase. Le terme dépendant de g dans $R_{i,j,l}$ est donné par :

$$-\mu_s I_{i,j,l}^n(g, u) = -\mu_s \sum_{k=0}^{K-1} w_k u_{i,j,k}^n \Phi_g(v_l, v_k)$$

Nous devons calculer pour chaque paire (l, k) la dérivée $\frac{\partial}{\partial g} \Phi_g(v_l, v_k)$.

Avec

$$\Phi_g(v_l, v_k) = \frac{1}{2\pi} \frac{N(g)}{D_{l,k}(g)}$$

, où $N(g) = 1 - g^2$ et $D_{l,k}(g) = 1 + g^2 - 2g(v_l \cdot v_k)$.

On procède par la règle du quotient :

(a) Pour le numérateur $N(g)$:

$$\frac{\partial N(g)}{\partial g} = -2g$$

(b) Pour le dénominateur $D_{l,k}(g)$:

$$\frac{\partial D_{l,k}(g)}{\partial g} = 2g - 2(v_l \cdot v_k)$$

Ainsi, en tenant compte du facteur $\frac{1}{2\pi}$, on obtient :

$$\frac{\partial}{\partial g} \Phi_g(v_l, v_k) = \frac{1}{2\pi} \left(\frac{-2g D_{l,k}(g) - 2N(g) \frac{\partial D_{l,k}(g)}{\partial g}}{D_{l,k}(g)^2} \right)$$

En remplaçant les dérivées, on obtient :

$$\frac{\partial}{\partial g} \Phi_g(v_l, v_k) = \frac{1}{2\pi} \left(\frac{-2gD_{l,k}(g) - 2N(g)(2g - 2(v_l \cdot v_k))}{D_{l,k}(g)^2} \right)$$

En factorisant par -2 , on a :

$$\frac{\partial}{\partial g} \Phi_g(v_l, v_k) = -\frac{1}{\pi} \left(\frac{gD_{l,k}(g) + N(g)(g - (v_l \cdot v_k))}{D_{l,k}(g)^2} \right)$$

Donc la dérivée totale par rapport à g est donnée par :

$$\begin{aligned} \frac{\partial R_{i,j,l}}{\partial g} &= -\mu_s \sum_{k=0}^{K-1} w_k u_{i,j,k}^n \left(-\frac{1}{\pi} \left(\frac{gD_{l,k}(g) + N(g)(g - (v_l \cdot v_k))}{D_{l,k}(g)^2} \right) \right) \\ &= \frac{\mu_s}{\pi} \sum_{k=0}^{K-1} w_k u_{i,j,k}^n \left(\frac{gD_{l,k}(g) + N(g)(g - (v_l \cdot v_k))}{D_{l,k}(g)^2} \right) \\ &= \frac{\mu_s}{\pi} \sum_{k=0}^{K-1} w_k u_{i,j,k}^n \left(\frac{g(1 + g^2 - 2g(v_l \cdot v_k)) + (1 - g^2)(g - (v_l \cdot v_k))}{(1 + g^2 - 2g(v_l \cdot v_k))^2} \right) \end{aligned}$$

4. **Conclusion** En somme on a les dérivées suivantes :

Dérivée	Expression
$\frac{\partial R_{i,j,l}}{\partial u_{i,j,l}^n}$	$-\frac{1}{\delta t} + \frac{v_{l,x}}{\Delta x} + \frac{v_{l,y}}{\Delta y} + (\mu_a + \mu_s) - \mu_s w_l \Phi_g(v_l, v_l)$
$\frac{\partial R_{i,j,l}}{\partial u_{up(i),j,l}^n}$	$-\frac{v_{l,x}}{\Delta x}$
$\frac{\partial R_{i,j,l}}{\partial u_{i,up(j),l}^n}$	$-\frac{v_{l,y}}{\Delta y}$
$\frac{\partial R_{i,j,l}}{\partial u_{i,j,k}^n} (k \neq l)$	$-\mu_s w_k \Phi_g(v_l, v_k)$
$\frac{\partial R_{i,j,l}}{\partial \mu_a}$	$u_{i,j,l}^n$
$\frac{\partial R_{i,j,l}}{\partial \mu_s}$	$u_{i,j,l}^n - I_{i,j,l}^n(g, u)$
$\frac{\partial R_{i,j,l}}{\partial g}$	$\frac{\mu_s}{\pi} \sum_{k=0}^{K-1} w_k u_{i,j,k}^n \left(\frac{g D_{l,k}(g) + N(g)(g - (v_l \cdot v_k))}{D_{l,k}(g)^2} \right)$

TABLE 1 – Tableau récapitulatif des dérivées de $R_{i,j,l}$

Ainsi, tout en gardant la dérivation formelle pour la pédagogie, nous nous appuyons sur PyTorch pour garantir que la matrice F corresponde exactement à l'implémentation numérique du schéma.