

6. MÉTHODES SÉQUENTIELLES : FILTRE DE KALMAN

Le filtre de Kalman (KF) et sa version étendue (EKF, Extended Kalman Filter) sont des extensions de l'analyse moindres carrés dans le cas de l'assimilation de données séquentielle, dans laquelle chaque ébauche est fournie par une prévision issue de l'analyse faite sur la fenêtre d'assimilation précédente. Les équations du filtre de Kalman linéaire sont les mêmes que celles décrites dans l'analyse moindres carrés. On changera certaines notations : l'ébauche est désormais notée x_f car c'est une prévision issue de la fenêtre précédente, et les matrices de covariance d'erreur d'analyse et d'ébauche (prévision) sont désormais notées P^a et P^f respectivement.

6.1. Notations et hypothèses. Les notations et hypothèses sont essentiellement les mêmes que pour l'analyse moindres carrés, ou pour le 4D-VAR, on précise juste les choses suivantes :

- Les matrices de covariance d'erreur d'ébauche et d'analyse sont notées P^f (forecast, prévision) et P^a (analysis, analyse) respectivement. L'ébauche est désormais vue comme une prévision (éventuellement provenant d'une période temporelle passée).
- L'opérateur de modèle (la résolvante) permettant de passer de l'instant t_{i-1} à l'instant t_i est noté M_i .
- Quand elle est modélisée, l'erreur de modèle est l'écart entre l'état réel du système $x_t(t_i)$ à l'instant t_i et la prévision déduite de l'état réel à l'instant passé (propagé par le modèle) : $M_i(x_t(t_{i-1}))$, et on suppose que cette erreur est non biaisée, et on note Q_i sa matrice de covariance.
- Les erreurs de modèle et d'analyse sont décorréliées.
- L'opérateur de modèle peut être supposé linéaire, ou linéarisé autour de l'analyse : $M_i(x(t_{i-1})) - M_i(x_a(t_{i-1})) = [M'_i(x_a)](x(t_{i-1}) - x_a(t_{i-1}))$.

6.2. Filtre de Kalman. Sous les hypothèses précédentes, on a le résultat suivant :

Théorème 6.1. *La façon optimale (au sens des moindres carrés) d'assimiler séquentiellement les observations est donnée par le filtre de Kalman (Kalman filter, KF), défini par récurrence sur les instants d'observation t_i :*

- *Étape de prévision* : $x_f(t_i) = M_i x_a(t_{i-1})$
- *Covariance d'erreur de prévision* : $P_i^f = M_i P_{i-1}^a M_i^T + Q_i$
- *Matrice de gain de Kalman* : $K_i = P_i^f H_i^T (H_i P_i^f H_i^T + R_i)^{-1}$
- *Étape d'analyse* : $x_a(t_i) = x_f(t_i) + K_i (y_i - H_i x_f(t_i))$
- *Covariance d'erreur d'analyse* : $P_i^a = (I - K_i H_i) P_i^f$

Démonstration. L'étape de prévision prévoie juste qu'on ne peut utiliser que le modèle pour faire évoluer l'état entre les instants t_{i-1} et t_i , et on le fait en partant du précédent état optimal, à savoir l'analyse $x_a(t_{i-1})$.

On doit ensuite calculer l'erreur de prévision :

$$\begin{aligned} x_f(t_i) - x_t(t_i) &= M_i(x_a(t_{i-1})) - M_i(x_t(t_{i-1})) + M_i(x_t(t_{i-1})) - x_t(t_i) \\ &= M_i(x_a(t_{i-1}) - x_t(t_{i-1})) + M_i(x_t(t_{i-1})) - x_t(t_i) \end{aligned}$$

et l'erreur de prévision se décompose donc comme la somme de l'erreur d'analyse à l'instant t_{i-1} propagée par le modèle M_i à l'instant t_i , et de l'erreur de modèle à l'instant t_i . Par décorrélation des deux erreurs, la covariance d'erreur sera la somme des deux covariances, la deuxième étant la matrice Q_i . Pour la première, on obtient

par définition de la covariance l'espérance de $M_i(x_a(t_{i-1}) - x_t(t_{i-1}))(x_a(t_{i-1}) - x_t(t_{i-1}))^T M_i^T$ qui vaut $M_i P_{i-1}^a M_i^T$.

Enfin, le gain optimal et l'analyse optimale (et sa matrice de covariance d'erreur) sont exactement celles définies par l'analyse moindres carrés (comme tout se passe à un instant fixé, t_i). \square

6.3. Équivalence avec le 4D-VAR. On a le résultat d'équivalence suivant :

Théorème 6.2. *Sur la même période d'assimilation, si on suppose que le modèle est parfait ($Q = 0$), et que les deux algorithmes utilisent les mêmes données (observations, ébauche de la condition initiale, et matrice de covariance d'erreur d'ébauche à l'instant initial), alors il y a égalité entre :*

- (1) l'analyse finale $x_a(t_N)$ produite par le filtre de Kalman, et
- (2) la valeur finale de la trajectoire optimale estimée par le 4D-VAR, c'est-à-dire $M_{0 \rightarrow t_N} x_0^*$.

Il est remarquable de noter qu'on arrive au même résultat alors que les approches sont totalement différentes. Dans un cas, à chaque nouvelle observation, on réalise une analyse optimale (tri-dimensionnelle, sans considération du temps), alors que dans l'autre cas, on cherche un minimum global du problème quadri-dimensionnel (on minimise l'écart global, sur toute la trajectoire en temps).

6.4. Filtre de Kalman étendu. Lorsque le modèle n'est pas linéaire, et/ou que l'opérateur d'observation H n'est pas linéaire, les opérateurs sont linéarisés (ou dérivés) autour de l'état courant, typiquement la dernière analyse ou prévision, et les mêmes formules s'appliquent, avec les opérateurs linéarisés pour la définition de la matrice de gain, et pour la propagation des matrices de covariance.

6.5. Commentaires. Le coût de mise en œuvre du filtre de Kalman (étendu ou non) est essentiellement le coût de calcul de l'analyse et de sa matrice de covariance d'erreur. La propagation de la matrice de covariance requiert n résolution du modèle (linéaire tangent et/ou adjoint), pour propager chacun des n vecteurs-colonnes de l'ancienne matrice. Les coûts de stockage (des matrices de covariance, notamment) et de calcul sont donc largement supérieurs à la mise en œuvre du 4D-VAR, même pour des petits modèles.

De nombreux filtres ont été développés ces dernières années, notamment en utilisant la réduction d'ordre (décomposition des matrices en valeurs singulières : POD, SVD, ...), pour s'affranchir de cette difficulté, et se restreindre (comme dans l'interpolation optimale) à des dimensions plus raisonnables, en ne conservant que des sous-matrices des matrices complètes.

Exemple du filtre SEEK : on suppose que la matrice P_0^a est décomposée sous la forme $S_0^a S_0^{aT}$ où S est une matrice $r \times n$, avec $r \ll n$ (par exemple en tronquant une SVD sur la matrice P). Les formules de mise à jour deviennent :

- Chaque vecteur colonne de S^a est propagé pour en déduire la colonne correspondante de S^f : $[S_i^f]_j = M_i(x_a(t_{i-1}) + [S_{i-1}^a]_j) - M_i(x_a(t_{i-1}))$;
- Matrice de gain : $K_i = S_i^f [I + (H S_i^f)^T R^{-1} (H S_i^f)]^{-1} (H S_i^f)^T R^{-1}$;
- Covariance d'erreur d'analyse : $P_i^a = S_i^f [I + (H S_i^f)^T R^{-1} (H S_i^f)]^{-1} S_i^{fT}$, décomposable sous la forme $S_i^a S_i^{aT}$ avec une SVD de la matrice à l'intérieur du crochet.

Le coût du filtre SEEK devient simplement le coût de propagation des r vecteurs de la décomposition des matrices de covariance d'erreur.