المدرسة الحسنية للأشغال العمومية
ECOLE HASSANIA DES TRAVAUX PUBLICS

Numerical Modelling Projects, Meteorology 2020/2021

4th project: Variational bias correction inside 4D-Var : Application to a simplified problem

Réalisé par :

CHAQDID Abdelaziz

3ème année génie Météorologie

Ecole Hassania des Travaux Publics (EHTP)

Encadré par :

Pr. Noureddine Semane

Ecole Hassania des Travaux Publics (EHTP)

Le 16 Décembre 2020

Model formulation

Soit le modèle suivant :

$$\frac{\partial q}{\partial t} + u\frac{\partial q}{\partial \lambda} = 0$$

$$\frac{\partial u}{\partial t} = 0$$

$$\frac{\partial u}{\partial \lambda} = 0$$

$$q\,(2\pi, t) = q(0, t)$$

$$q\,(\lambda, \mathbf{0}) = q_0(\lambda)$$

u est invariant dans le temps et espace.

(λ ,t)$\in$ $[0, 2\pi] *]0, T]$

$x\,(t) = \binom{q(\lambda,t)}{u}$ vecteur d'état

Pourvu de simplifier, on prend $\Delta t = \Delta\lambda$

La grille du modèle est constituée de 3 longitudes discrètes $(\lambda_j,\ j = 1, 2, 3)$, d'où le vecteur d'état discrétisé : $x_j^n = \binom{q(\lambda_j,t_n)}{u} = \binom{q_j^n}{u}$

the discretized equation of the model :

$$\frac{\partial q}{\partial t} + u\frac{\partial q}{\partial \lambda} = 0 \ (1)$$

Pour calculer la dérivée spatiale de façon discrète, on se donne 3 points de grille avec une discrétisation $_{\mathrm{j}}$= (j − 1)Δ avec Δ=$\frac{2}{3}$ et j= 1,2,3. On utilisera la différentiation centrée d'ordre 2.

Pour le calcul de la dérivée temporelle, on se donne une discrétisation $t_n = n\Delta t$ avec 0≤nΔt≤T, et on applique la méthode des différences finies :

Discrétisation de $\frac{\partial q}{\partial t}$ à l'aide de la formule des différences finies avant d'ordre 1.

$$\frac{\partial q}{\partial t}(\lambda_j, t_n) = \frac{Q_j^{n+1} - Q_j^n}{\Delta t} \quad \text{(first order forward differencing)} \quad (2)$$

Discrétisation de$\frac{\partial q}{\partial \lambda}$ à l'aide de la formule des différences finies centrées d'ordre 2.

$$\frac{\partial q}{\partial \lambda}(\lambda_j, t_n) = \frac{Q_{j+1}^n - Q_{j-1}^n}{2\Delta\lambda} \quad \text{(second order central differencing)} \quad (3)$$

En substituant (2) et (3) dans (1), on obtient :

$$\frac{Q_j^{n+1} - Q_j^n}{\Delta t} = -u\frac{Q_{j+1}^n - Q_{j-1}^n}{2\Delta\lambda}$$

$$Q_j^{n+1} - Q_j^n = -\frac{u}{2}(Q_{j+1}^n - Q_{j-1}^n)$$

Où $\Delta t = \Delta\lambda$ **,**

D'où,

$$\boldsymbol{Q_j^{n+1} = Q_j^n - \frac{u}{2}(Q_{j+1}^n - Q_{j-1}^n)} \ (4)$$

1 the discretized model in matrix form :

Sachant que le vent est certain, le nouveau vecteur d'état est $x^n = q(\lambda_j, t_n)$, qui s'écrit dans le cas discrétisé comme suit : $x^n = \begin{pmatrix} Q_1^n \\ Q_2^n \\ Q_3^n \end{pmatrix}$

l'évaluation de l'équation (4) dans les points de grille j=1, 2 et 3 donne :

$$\begin{cases} Q_1^{n+1} = Q_1^n & - \frac{u}{2}Q_2^n + \frac{u}{2}Q_0^n \\ Q_2^{n+1} = Q_2^n & - \frac{u}{2}Q_3^n + \frac{u}{2}Q_1^n \\ Q_3^{n+1} = Q_3^n & - \frac{u}{2}Q_4^n + \frac{u}{2}Q_2^n \end{cases}$$

Or, le champ traceur est périodique dans l'espace, donc,

$$\begin{cases} Q_1^{n+1} = Q_1^n & - \frac{u}{2}Q_2^n + \frac{u}{2}Q_3^n \\ Q_2^{n+1} = Q_2^n & - \frac{u}{2}Q_3^n + \frac{u}{2}Q_1^n \\ Q_3^{n+1} = Q_3^n & - \frac{u}{2}Q_1^n + \frac{u}{2}Q_2^n \end{cases}$$

Ce système peut s'écrire sous la forme matricielle comme suit :

$$\boldsymbol{x^{n+1}} = \mathcal{M}\boldsymbol{x^n}$$

Avec, $x^{n+1} = \begin{pmatrix} Q_1^{n+1} \\ Q_2^{n+1} \\ Q_3^{n+1} \end{pmatrix}$, $x^n = \begin{pmatrix} Q_1^n \\ Q_2^n \\ Q_3^n \end{pmatrix}$ et $\mathcal{M} = \begin{pmatrix} 1 & -\frac{u}{2} & +\frac{u}{2} \\ +\frac{u}{2} & 1 & -\frac{u}{2} \\ -\frac{u}{2} & +\frac{u}{2} & 1 \end{pmatrix}$

Ou encore, si on prend en considération le biais des observations :

$$\boldsymbol{x^{n+1}} = \widetilde{\mathcal{M}}\boldsymbol{x^n}$$

Avec, $x^{n+1} = \begin{pmatrix} Q_1^{n+1} \\ Q_2^{n+1} \\ Q_3^{n+1} \\ \beta \end{pmatrix}$, $x^n = \begin{pmatrix} Q_1^n \\ Q_2^n \\ Q_3^n \\ \beta \end{pmatrix}$ et $\widetilde{\mathcal{M}} = \begin{pmatrix} 1 & -\frac{u}{2} & +\frac{u}{2} & 0 \\ +\frac{u}{2} & 1 & -\frac{u}{2} & 0 \\ -\frac{u}{2} & +\frac{u}{2} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

2 tangent linear model equation.

On introduit une perturbation $\hat{q}$ sur le vecteur d'état à l'instant initial t=0
Donc, $\tilde{q} = q_b + \hat{q}$ où q_b et l'ébauche initiale du traceur.
Les équations vérifiées par les deux systèmes, initial et perturbé , sont :

$$\begin{cases} u\frac{\partial q_b}{\partial \lambda} = 0 \\ q_b\left(2\pi, t\right) = q_b(0, t) \\ q_b = q_0 \end{cases} \quad et \quad \begin{cases} \frac{\partial \tilde{q}}{\partial t} + u\frac{\partial \tilde{q}}{\partial \lambda} = 0 \\ \tilde{q}\left(2\pi, t\right) = \tilde{q}(0, t) \\ \tilde{q}\left(\lambda, 0\right) = q_b + \ \hat{q} \end{cases}$$

En soustrayant les deux systèmes, on obtient :

$$\begin{cases} \frac{\partial(q_b + h\hat{q})}{\partial t} + u\frac{\partial(q_b + h\hat{q})}{\partial \lambda} - u\frac{\partial q_b}{\partial \lambda} = 0 \\ \tilde{q}\left(2\pi, t\right) - \ q_b\left(2\pi, t\right) = \tilde{q}(0, t) - \ q_b(0, t) \\ \tilde{q}\left(\lambda, 0\right) - \ q_b = h\hat{q} \end{cases}$$

$$\begin{cases} h\frac{\partial \hat{q}}{\partial t} + uh\frac{\partial \hat{q}}{\partial \lambda} = 0 \\ h\hat{q}\left(2\pi, t\right) = h\hat{q}(0,t) \\ h\hat{q}\left(\lambda, 0\right) = h\hat{q} \end{cases}$$

En divisant par h et la tendre vers 0, on obtient les équations du linéaire tangent :

$$\begin{cases} \boldsymbol{\frac{\partial \widehat{q}}{\partial t} + u\frac{\partial \widehat{q}}{\partial \lambda} = 0} \\ \boldsymbol{\widehat{q}\left(2\pi, t\right) = \widehat{q}(0,t)} \\ \boldsymbol{\widehat{q}\left(\lambda, 0\right) = \widehat{q}} \end{cases} \qquad \textbf{(6)}$$

3 *the discretized equation of the tangent linear model :*

Idem pour cette étape, le modèle linéaire tangent va être discrétisé en appliquant la formule des différences finies avant d'ordre 1 pour la dérivée temporelle, et la formule des différences finies centrées d'ordre 2 pour la dérivée spatiale.
Donc on a
$\frac{\partial \hat{q}}{\partial t}\left(\lambda_j, t_n\right) = \frac{\hat{q}_j^{n+1} - \hat{q}_j^n}{\Delta t}$ et $\frac{\partial \hat{q}}{\partial \lambda}(\lambda_j, t_n) = \frac{\hat{q}_{j+1}^n - \hat{q}_{j-1}^n}{2\Delta\lambda}$
En injectant dans l'équation (**??**), on obtient ce qui suit :

$$\hat{q}_j^{n+1} - \hat{q}_j^n + \frac{u}{2}\left(\hat{q}_{j+1}^n - \hat{q}_{j-1}^n\right) = 0$$

D'où le modèle linéaire tangent discrétisé :

$$\boldsymbol{\widehat{q}_j^{n+1} = \widehat{q}_j^n - \frac{u}{2}\widehat{q}_{j+1}^n + \frac{u}{2}\widehat{q}_{j-1}^n}$$

4 *the discretized tangent linear model in matrix form*

le modèle linéaire tangent discrétisé est donnée par :$\hat{q}_j^{n+1} = \hat{q}_j^n - \frac{u}{2}\hat{q}_{j+1}^n + \frac{u}{2}\hat{q}_{j-1}^n$
donc,

$$\begin{cases} \hat{q}_1^{n+1} = \hat{q}_1^n - \frac{u}{2}\hat{q}_2^n + \frac{u}{2}\hat{q}_0^n \\ \hat{q}_2^{n+1} = \hat{q}_2^n - \frac{u}{2}\hat{q}_3^n + \frac{u}{2}\hat{q}_1^n \\ \hat{q}_3^{n+1} = \hat{q}_3^n - \frac{u}{2}\hat{q}_4^n + \frac{u}{2}\hat{q}_2^n \end{cases}$$

Or, le champ traceur est périodique dans l'espace, donc,

$$\begin{cases} \hat{q}_1^1 = \hat{q}_1^0 - \frac{u}{2}\hat{q}_2^0 + \frac{u}{2}\hat{q}_3^0 \\ \hat{q}_2^1 = \hat{q}_2^0 - \frac{u}{2}\hat{q}_3^0 + \frac{u}{2}\hat{q}_1^0 \\ \hat{q}_3^1 = \hat{q}_3^0 - \frac{u}{2}\hat{q}_1^0 + \frac{u}{2}\hat{q}_2^0 \end{cases}$$

Ce système peut s'écrire sous la forme matricielle comme suit :

$$\boldsymbol{\widehat{x}\left(t_0 + \Delta t\right)} = \mathbf{M}\boldsymbol{\widehat{x}\left(t_0\right)} \qquad \textbf{(7)}$$

Où $\widehat{\boldsymbol{x}}\left(t_0 + \Delta t\right) = \begin{pmatrix} \hat{q}_1\left(t_0 + \Delta t\right) \\ \hat{q}_2\left(t_0 + \Delta t\right) \\ \hat{q}_3\left(t_0 + \Delta t\right) \end{pmatrix}$, $\widehat{\boldsymbol{x}}\left(t_0\right) = \begin{pmatrix} \hat{q}_1\left(t_0\right) \\ \hat{q}_2\left(t_0\right) \\ \hat{q}_3\left(t_0\right) \end{pmatrix}$

Et $\mathbf{M} = \begin{pmatrix} 1 & -\frac{u_b}{2} & +\frac{u_b}{2} \\ +\frac{u_b}{2} & 1 & -\frac{u_b}{2} \\ -\frac{u_b}{2} & +\frac{u_b}{2} & 1 \end{pmatrix}$

5 the adjoint model in matrix form

$$\mathrm{M}^{*}=\mathrm{M}^{\mathrm{T}}=\mathrm{M}=\begin{pmatrix} \mathbf{1} & +\frac{u_b}{2} & -\frac{u_b}{2} \\ -\frac{u_b}{2} & \mathbf{1} & +\frac{u_b}{2} \\ +\frac{u_b}{2} & -\frac{u_b}{2} & \mathbf{1} \end{pmatrix}$$

6 the value of the observation operator

on observe le traceur sur tous les points de grille alors l'opérateur d'observation s'écrit :

$$\boldsymbol{H}=\begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$$

Ou encore, si on prend en considération le biais des observations :

$$\widetilde{\mathrm{H}}=\begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \end{pmatrix}$$

7 value of the tracer observation minus forecast departure

le vecteur d'innovation s'écrit **:**

$$d_1 = q_{obs} - \ Hx^1{}_b$$

Or, $q_{obs} = \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{M}}\,(x_t(0))$
Donc,

$$\boldsymbol{d_1} = \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{M}}\left(\boldsymbol{x_t^0} - \boldsymbol{x_b^0}\right)$$

où,

$$x_t(0) = \begin{pmatrix} q_t{}^0_1 \\ q_t{}^0_2 \\ q_t{}^0_3 \\ \beta_t \end{pmatrix} \text{ et } x_b(0) = \begin{pmatrix} q_b{}^0_1 \\ q_b{}^0_2 \\ q_b{}^0_3 \\ \beta_b = 0 \end{pmatrix}$$

8 the value of the 4D-Var analysis at the initial time

l'expression de l'analyse en 4D-var est donnée par :

à t_0 :

$$x^0{}_a = x^0{}_b + K_0 d_1$$

où $K_0 = B(\widetilde{H}\widetilde{M})^T \left((\widetilde{H}\widetilde{M})B(\widetilde{H}\widetilde{M})^T + R\right)^{-1}$

à t_1 :

$$x^1{}_a = x^1{}_b + K_1 d_1$$

où $K_1 = B\widetilde{H}^T \left(\widetilde{H}B\widetilde{H}^T + R\right)^{-1}$

9 *the weak constraint 4D-Var ?*

la fonction coût est donnée par :

$$J(x_0,\ x_1) = \tfrac{1}{2}\left[x(0) - x_b(0)\right]^{T\mathbf{B^{-1}}}\left[x(0) - x_b(0)\right]$$
$$+\frac{1}{2}\left[\widetilde{H}(x(t_1)) - q_{obs}\right]^T \mathbf{R^{-1}}\left[\widetilde{H}(x(t_1)) - q_{obs}\right] + \frac{1}{2}\eta^T\mathbf{Q^{-1}}\eta$$

où $\eta = q(t_1) - M(q(0))$

Soit $\widetilde{\eta}$ l'erreur du modèle biaisée, telle que :

$$\widetilde{\eta} = x(t_1) - \widetilde{M}(x(0)) = \delta x_1 - \widetilde{M}\delta x_0$$

Sachant que $x_b(t_1) - M(x_b(0)) = 0$, et posons $\delta x_0 = x(0) - x_b(0)$, la fonction coût s'écrit :

$$\mathbf{J}(\mathbf{x_0}, \mathbf{x_1}) = \frac{\mathbf{1}}{\mathbf{2}}\delta\mathbf{x_0}^{\mathbf{T}}\mathbf{B}^{-1}\delta\mathbf{x_0} + \frac{\mathbf{1}}{\mathbf{2}}\left[\widetilde{\mathbf{H}}\delta\mathbf{x_1} - \mathbf{d_1}\right]^{\mathbf{T}}\mathbf{R}^{-1}\left[\widetilde{\mathbf{H}}\delta\mathbf{x_1} - \mathbf{d_1}\right] + \frac{\mathbf{1}}{\mathbf{2}}\widetilde{\eta}^{\mathbf{T}}\mathbf{Q}^{-1}\widetilde{\eta}$$

observation

$$q_{obs} = \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{M}}_{\boldsymbol{t}}(x_t(0))$$

$$\text{Où } \widetilde{\boldsymbol{M}}_{\boldsymbol{t}} = \begin{pmatrix} 1-\ 2K & K-\frac{u}{2} & K+\frac{u}{2} & 0 \\ K+\frac{u}{2} & 1-\ 2K & K-\frac{u}{2} & 0 \\ K-\frac{u}{2} & K+\frac{u}{2} & 1-\ 2K & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

innovation

$$d_1 = q_{obs} - \tilde{H}x^1{}_b$$

Or, $x^1{}_b = \widetilde{\boldsymbol{M}}(x^0{}_b)$ *et* $q_{obs} = \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{M}}_{\boldsymbol{t}}(x_t(0))$

Analyse à t1

l'expression de l'analyse t_1 en 4D-var est donnée par :$\mathrm{x}^1{}_{\mathrm{a}} = \mathrm{x}^1{}_{\mathrm{b}} + \mathrm{x}^{\mathrm{a}}_{\mathrm{t}_1}$

où

$$\mathrm{x}_{\mathrm{t}_1}^{\mathrm{a}} = \left(\widetilde{\mathrm{M}}\mathrm{B}\widetilde{\mathrm{M}}^{\mathrm{T}} + \mathrm{Q}_\beta\right)\tilde{\mathrm{H}}^{\mathrm{T}}\left(\tilde{\mathrm{H}}\left(\widetilde{\mathrm{M}}\mathrm{B}\widetilde{\mathrm{M}}^{\mathrm{T}} + \mathrm{Q}_\beta\right)\tilde{\mathrm{H}}^{\mathrm{T}} + \mathrm{R}\right)^{-1}\mathrm{d}_1$$

Analyse à t0

Et $x^0{}_a = x^0{}_b + \delta x^a_{t_0}$
où

$$\delta x^a_0 = B\tilde{M}^T\tilde{H}^T\left(\tilde{H}\left(\tilde{M}B\tilde{M}^T + Q_\beta\right)\tilde{H}^T + R\right)^{-1} d_1$$

Code Python

4rd project: Variational bias correction inside 4D-Var : Application to a simplified problem

importation des packages

In [1]:
```
import numpy as np
import math
from numpy.linalg import inv
```

La contrainte forte

définition des variables

In [2]:
```
sig2_obs =  0.001* 0.001   # variance d'erreur d'obs
sig2_q = 1*1         # variance d'erreur de q
sig2_b = sig2_q      # variance d'erreur du biais
beta = 0.5              # le biais
u = 1                # le vent
print('\nthe wind u : ',np.round(u,4),
      '\nbaias beta : ',np.round(beta,4))

xt = np.array([[2.5, 3.4, 1.3, beta]]).transpose()  # vecteur de l'état réelle à t=0
print('\ntruth vector at t0 :\n',np.round(xt,4))

xb_0 = np.array([[2, 3, 1, 0]]).transpose()  # l'ébauche à t=0
print('\nBackgroung vector at t0 :\n',np.round(xb_0,4))
```

```
the wind u :  1
baias beta :  0.5

truth vector at t0 :
 [[2.5]
 [3.4]
 [1.3]
 [0.5]]

Backgroung vector at t0 :
 [[2]
 [3]
 [1]
 [0]]
```

Matrice d'erreurs d'observation

$$R = \begin{pmatrix} {\sigma^2}_{obs} & 0 & 0 \\ 0 & {\sigma^2}_{obs} & 0 \\ 0 & 0 & {\sigma^2}_{obs} \end{pmatrix}$$

In [3]:
```
R = np.array([[sig2_obs, 0, 0 ],
              [0, sig2_obs, 0 ],
              [0, 0, sig2_obs ]]) # matrice de covariance des erreurs d'obs
print('\nObsevation error variance matrix :\n',np.round(R,8))
```

```
Obsevation error variance matrix :
 [[1.e-06 0.e+00 0.e+00]
 [0.e+00 1.e-06 0.e+00]
 [0.e+00 0.e+00 1.e-06]]
```

Matrice d'erreurs d'ébauche

$$B = \begin{pmatrix} \sigma^2{}_q & 0 & 0 & 0 \\ 0 & \sigma^2{}_q & 0 & 0 \\ 0 & 0 & \sigma^2{}_q & 0 \\ 0 & 0 & 0 & \sigma^2{}_\beta \end{pmatrix}$$

In [4]:
```
B = np.array([[sig2_q, 0, 0, 0 ],
              [0, sig2_q, 0, 0 ],
              [0, 0, sig2_q, 0 ],
              [0, 0, 0, sig2_b ]]) # matrice de covariance des erreurs d'ébauche
print('\nBackground error variance matrix :\n',np.round(B,4))
```

```
Background error variance matrix :
 [[1 0 0 0]
 [0 1 0 0]
 [0 0 1 0]
 [0 0 0 1]]
```

Operateur d'observation avec biais

$$\tilde{H} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

In [5]:
```
H = np.array([[1, 0, 0, 1],
              [0, 1, 0, 1],
              [0, 0, 1, 1]]) # Operateur d'obs
print('\nobs operator :\n',np.round(H,4))
```

```
obs operator :
 [[1 0 0 1]
 [0 1 0 1]
 [0 0 1 1]]
```

Matrice du modèle imparfait avec biais

$$\tilde{M} = \begin{pmatrix} 1 & -\frac{u_b}{2} & +\frac{u_b}{2} & 0 \\ +\frac{u_b}{2} & 1 & -\frac{u_b}{2} & 0 \\ -\frac{u_b}{2} & +\frac{u_b}{2} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In [6]:
```
M = np.array([[ 1    , -u/2 , +u/2, 0],
              [ +u/2 , 1    , -u/2, 0],
              [ -u/2 , +u/2 , 1   , 0],
              [ 0    , 0    , 0   , 1]]) # matrice du modèle imparfait
print('\nimperfect model : \n',np.round(M,4))
```

```
imperfect model :
 [[ 1.  -0.5  0.5  0. ]
 [ 0.5  1.  -0.5  0. ]
 [-0.5  0.5  1.   0. ]
 [ 0.   0.   0.   1. ]]
```

Vecteur d'observation

$$q_{obs} = \tilde{H}\tilde{M}(x^0{}_t)$$

In [7]:
```
y0=np.dot(H, M.dot(xt))  # obserrvation a t1
print('\nObservation at t1 : \n',np.round(y0,3))
```

```
Observation at t1 :
 [[1.95]
 [4.5 ]
 [2.25]]
```

Vecteur d'innovation

$$d_1 = q_{obs} - \tilde{H} x^1{}_b$$

In [8]:
```
d =y0-np.dot(np.dot(H,M),xb_0) # inovation a t1
print('\nInnovation at t1 : \n',np.round(d,3))
```

```
Innovation at t1 :
 [[0.95]
 [1.  ]
 [0.75]]
```

Analyse à t_0

$$x^0{}_a = x^0{}_b + K_0 d_1$$

où

$$K_0 = B(\tilde{H}\tilde{M})^T \left((\tilde{H}\tilde{M})B(\tilde{H}\tilde{M})^T + R\right)^{-1}$$

In [9]:
```
HM=np.dot(H,M)
HMT=np.transpose(HM)
BHMT=np.dot(B,HMT)
HMBHMT=np.dot(HM,BHMT)
K_0=np.dot(BHMT,inv(HMBHMT+R))
xa_0= xb_0 + np.dot(K_0,d)
print('\nAnalysis at t0 : \n',np.round(xa_0,3))
```

```
Analysis at t0 :
 [[2.325]
 [3.225]
 [1.125]
 [0.675]]
```

Analyse à t_1

$$x^1{}_a = x^1{}_b + K_1 d_1$$

où

$$K_1 = B\tilde{H}^T \left(\tilde{H}B\tilde{H}^T + R\right)^{-1}$$

In [10]:
```
HT=np.transpose(H)
BHT=np.dot(B,HT)
HBHT=np.dot(H,BHT)
K_1=np.dot(BHT,inv(HBHT+R))
xb_1 = np.dot(M,xb_0)
xa_1= xb_1 + np.dot(K_1,d)
print('\nAnalysis at t1 : \n',np.round(xa_1,3))
```

```
Analysis at t1 :
 [[1.275]
 [3.825]
 [1.575]
 [0.675]]
```

La contrainte faible

définition des variables

In [11]:
```
sig2_obs =  0.001* 0.001   # variance d'erreur d'obs
sig2_q = 1*1         # variance d'erreur de q
sig2_b = sig2_q      # variance d'erreur du biais
sig2_m = sig2_q      # variance d'erreur du modèle
beta = 0.5           # le biais
u = 1                # le vent
k = 0.4              # le vent

print('\ndiffusion coef :', np.round(k,4),
      '\nthe wind u : ',np.round(u,4),
      '\nbaias beta : ',np.round(beta,4))

xt = np.array([[2.5, 3.4, 1.3, beta]]).transpose()  # vecteur de l'état réelle à t=0
print('\ntruth vector at t0 :\n',np.round(xt,4))

xb_0 = np.array([[2, 3, 1, 0]]).transpose()  # l'ébauche à t=0
print('\nBackgroung vector at t0 :\n',np.round(xb_0,4))
```

```
diffusion coef : 0.4
the wind u :  1
baias beta :  0.5

truth vector at t0 :
 [[2.5]
 [3.4]
 [1.3]
 [0.5]]

Backgroung vector at t0 :
 [[2]
 [3]
 [1]
 [0]]
```

Matrice d'erreurs du modèle

$$Q_\beta = \begin{pmatrix} {\sigma^2}_m & 0 & 0 & 0 \\ 0 & {\sigma^2}_m & 0 & 0 \\ 0 & 0 & {\sigma^2}_m & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

In [12]:
```
Q = np.array([[sig2_m, 0, 0, 0],
              [0, sig2_m, 0, 0],
              [0, 0, sig2_m, 0],
              [0, 0, 0      , 0]]) # matrice de covariance des erreurs du modèle
print('\nmodel error variance matrix :\n',np.round(Q,4))
```

```
model error variance matrix :
 [[1 0 0 0]
 [0 1 0 0]
 [0 0 1 0]
 [0 0 0 0]]
```

Matrice du modèle réel

$$M_t = \begin{pmatrix} 1-2K & K-\frac{u}{2} & K+\frac{u}{2} & 0 \\ K+\frac{u}{2} & 1-2K & K-\frac{u}{2} & 0 \\ K-\frac{u}{2} & K+\frac{u}{2} & 1-2K & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In [13]:

```
K=k*1.5/math.pi
Mt = np.array([[ 1-2*K , K-u/2 , K+u/2, 0],
               [ K+u/2 , 1-2*K , K-u/2, 0],
               [ K-u/2 , K+u/2 , 1-2*K, 0],
               [ 0     , 0     , 0    , 1]]) # matrice du  modèle vrai
print('\ntrue model :\n',np.round(Mt,4))
```

```
true model :
 [[ 0.618 -0.309  0.691  0.   ]
 [ 0.691  0.618 -0.309  0.   ]
 [-0.309  0.691  0.618  0.   ]
 [ 0.     0.     0.     1.   ]]
```

Vecteur d'observation

$$q_{obs} = \tilde{H}\tilde{M}_t({x^0}_t)$$

In [14]:

```
y0=np.dot(H, Mt.dot(xt))  # obserrvation a t1
print('\nObservation at t1 : \n',np.round(y0,3))
```

```
Observation at t1 :
 [[1.893]
 [3.927]
 [2.88 ]]
```

Vecteur d'innovation

$$d_1 = q_{obs} - \tilde{H}{x^1}_b$$

où

$${x^1}_b = \tilde{M}({x^0}_b)$$

In [15]:

```
d =y0-np.dot(np.dot(H,M),xb_0) # inovation a t1
print('\nInnovation at t1 : \n',np.round(d,3))
```

```
Innovation at t1 :
 [[0.893]
 [0.427]
 [1.38 ]]
```

Analyse à t_0

$${x^0}_a = {x^0}_b + \delta\, x^a_{t_0}$$

où

$$\delta\, x^a_0 = B\tilde{M}^T\tilde{H}^T\left(\tilde{H}\left(\tilde{M}B\tilde{M}^T + Q_\beta\right)\tilde{H}^T + R\right)^{-1} d_1$$

In [16]:
```
HT=H.transpose()
MT=M.transpose()
MB   = np.dot(M,B)
MBMT = np.dot(MB,MT)
X    = MBMT +Q
Y    = np.dot(H,X.dot(HT))
BMT  = np.dot(B,MT)
BMTHT= np.dot(BMT,HT)
delta_0= np.dot(BMTHT,inv(Y+R)).dot(d)

xa_0 = xb_0 + delta_0
print('\nAnalysis vector at t0 :\n',np.round(xa_0,3))
```

```
Analysis vector at t0 :
 [[2.004]
 [3.097]
 [1.439]
 [0.54 ]]
```

Analyseà t_1

$$x^1{}_a = x^1{}_b + \delta x^a_{t_1}$$

où

$$\delta x^a_{t_1} = \left(\tilde{M}B\tilde{M}^T + Q_\beta\right)\tilde{H}^T\left(\tilde{H}\left(\tilde{M}B\tilde{M}^T + Q_\beta\right)\tilde{H}^T + R\right)^{-1} d_1$$

In [17]:
```
XHT=np.dot(X,HT)
delta_1= np.dot(XHT,inv(Y+R)).dot(d)

xb_1   = M.dot(xb_0)

xa_1 = xb_1 + delta_1
print('\nAnalysis vector at t1 :\n',np.round(xa_1,3))
```

```
Analysis vector at t1 :
 [[1.353]
 [3.387]
 [2.34 ]
 [0.54 ]]
```