INRA
Unité de Recherche MIA
Domaine de Vilvert
78352 Jouy-en-Josas Cedex

Université Louis Pasteur
UFR de Mathématique
7 rue René Descartes
67084 Strasbourg Cedex

MODÉLISATION DYNAMIQUE DE LA HAUTEUR D'UNE MARE dans le cadre de l'étude épidémiologique de la Fièvre de la Vallée du Rift

Rapport de stage de deuxième année de Magistère

par Sophie PENISSON

Sous la direction de Christine JACOB, Directrice de Recherche

20 juin - 15 juillet 2005

REMERCIEMENTS

Merci tout particulièrement à Christine JACOB pour son accueil, sa disponibilité et son aide tout au long de mon stage.

Merci également à Alexandre SIBERT pour avoir, de Dakar, répondu à mes interrogations.

Merci enfin à Eric MONVERT, David LEGLAND et à l'ensemble de l'unité pour leur accueil chaleureux.

Table des matières

Chapitre 1

L'INRA et l'unité MIA du centre de Jouy-en-Josas

L'Institut National de la Recherche Agronomique est un établissement public à caractère scientifique et technologique, placé sous la double tutelle des ministères chargés de la Recherche et de l'Agriculture. Il est constitué de 14 départements de recherche touchant à l'agriculture, l'alimentation et l'environnement.

Le Centre de Recherche de Jouy-en-Josas (CRJ) est un centre INRA dont la recherche scientifique est principalement axée sur les aliments, les micro-organismes et les animaux.

Il abrite en outre l'unité MIA (Mathématiques et Informatique Appliquées) qui développe des recherches en statistiques, analyse des systèmes et analyse d'image.

Cette unité se compose de deux équipes :
- MathCell (analyse de systèmes biologiques au niveau cellulaire et tissulaire) ;
- MatRisq (évaluation de risques environnementaux, sanitaires ou épidémiologiques), dans le cadre de laquelle s'effectue mon stage.

L'un des thèmes de recherche de l'équipe MatRisq est la modélisation de l'émergence et de l'évolution d'épizooties. C'est à ce titre qu'elle s'est impliquée dans le projet Emercase, qui vise à l'alerte précoce de risques épizootiques ou épidémiques de maladies émergentes en Afrique.

Chapitre 2

Introduction du problème

2.1 La Fièvre de la Vallée du Rift

La Fièvre de la Vallée du Rift (FVR) a été choisie comme maladie générique du projet Emercase. C'est une maladie transmise par les moustiques qui touche les ruminants domestiques et sauvages, certains rongeurs et l'homme. Elle peut également être contagieuse pour l'homme par contact avec des animaux malades.

La mort de troupeaux touchés entraîne souvent des pertes économiques considérables, mais en raison de la nature de la maladie, la vaccination n'est pas la mesure préventive la plus efficace. Il paraît plus adapté, grâce à des modèles mathématiques appropriés, de chercher à déterminer le risque de survenue d'une flambée de FVR afin de prendre les mesures nécessaires.

Dans la région du Sénégal où l'étude sera effectuée, les moustiques vecteurs de cette maladie sont essentiellement du genre *Aedes* et *Culex*. Ces moustiques tiennent non seulement le rôle de *vecteurs* mais également de *réservoirs* car les femelles - qui transmettent le virus - restent infectantes toute leur vie.

Les moustiques sont généralement localisés à proximité des points d'eau. Le schéma suivant présente la localisation de leurs gîtes larvaires :

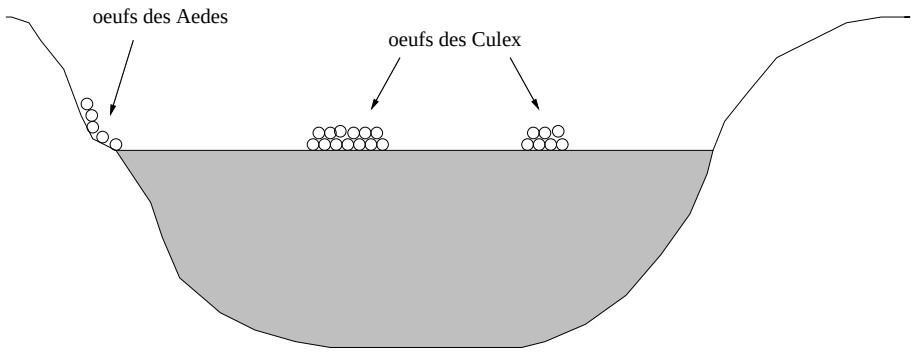


FIG. 2.1 – Localisation des gîtes larvaires

Les oeufs des *Culex* sont pondus sur la surface de la mare. Il y a des oeufs aussi longtemps qu'il y a de l'eau, et ils ne résistent pas à la sécheresse.

Les oeufs des *Aedes* sont pondus sur le rivage, au niveau de l'eau ou juste en dessous. Pour éclore, ils ont besoin d'une période d'assèchement et d'une mise en eau.

L'évolution des populations de moustiques est étroitement liée à celle des conditions climatiques telles que la température, le taux d'humidité, la pluviométrie et les vents.

On peut notamment schématiser la relation pluie/abondance de moustiques de la façon suivante :

- large mare profonde + pluies abondantes → pullulation de *Culex* ;
- large mare + pluies abondantes années précédentes → pullulation de *Aedes* l'année suivante.

On voit ainsi apparaître l'intérêt de connaître l'évolution de la hauteur des mares de la région étudiée pendant la saison des pluies.

2.2 Choix des mares étudiées

Le site de Barkédji, situé dans la région du Ferlo, au Sénégal, a été suivi pendant plusieurs années sur les plans entomologiques, virologiques et sérologiques. C'était donc un lieu idéal pour y tenter une simulation de la diffusion de la FVR.

Le choix des mares s'est porté sur deux mares à cuvette argileuse, de grande taille, qui restent en eau pendant toute la saison des pluies et une partie de la saison sèche :

- la mare de Barkédji, située dans le lit du Ferlo, en continuité avec d'autres mares du même lit et collectant l'eau de pluie sur une grande surface ;
- et la mare de Furdu, située hors du lit du Ferlo, remplie uniquement par son bassin versant.

L'étude relative à ce stage s'est portée uniquement sur la mare de Furdu, mais le modèle dynamique produit sera généralisable à d'autres mares.

Pour le site de Furdu, nous avions en notre possession plusieurs jeux de données :

- les relevés pluviométriques des années 2001 et 2002, relevés à Barkédji, et ceux de l'année 2003, relevés à la mare de Furdu même ;
- les relevés limnimétriques (hauteur d'eau) de la mare de Furdu des années 2001, 2002 et 2003 (plus ou moins fiables) ;
- enfin des données topographiques relatives à la mare de Furdu, notamment la relation surface d'eau/hauteur d'eau.

Chapitre 3

Modèle dynamique

3.1 Principe de la modélisation

L'objectif du modèle est de prédire l'évolution de la hauteur d'eau d'une mare, à partir des données de pluviométrie de ce lieu. Les relevés de hauteur de mare étant quotidiens, le pas de temps choisi sera de 24h.

Le modèle sera récursif, car la hauteur d'eau d'un jour donné dépend de celle de la journée précédente, mais il dépendra également de la pluviométrie, processus exogène. On prendra en compte les phénomènes physiques suivants :

- la pluie tombant directement sur la mare ;
- le ruissellement de l'eau sur les bords de la mare ;
- l'évaporation de l'eau ;
- l'infiltration de l'eau dans le sol sous la mare.

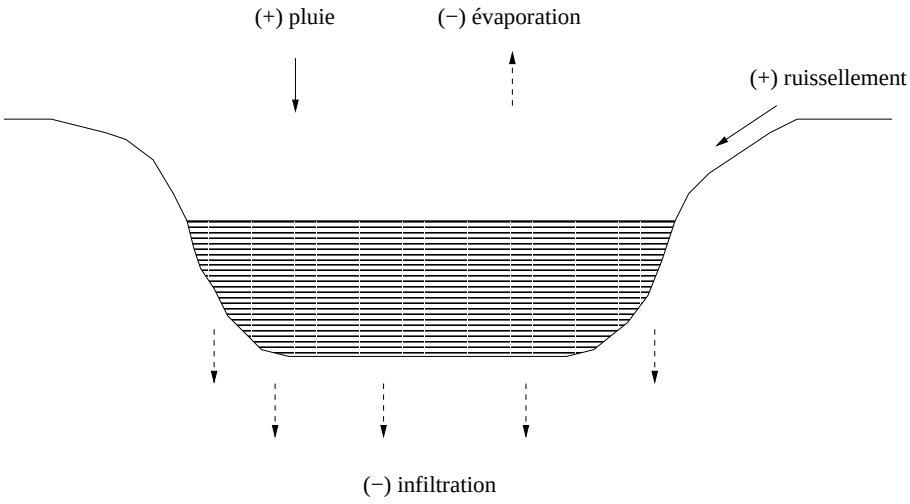


FIG. 3.1 – Phénomènes physiques

Le modèle a été conçu de manière à ce qu'il y ait le moins possible de paramètres à estimer, afin que l'incertitude soit minimale. Les paramètres à estimer seront essentiellement ceux relatifs à :

- la vitesse d'infiltration de l'eau dans le sol, qui sera liée à la porosité du sol et à sa saturation en eau ;
- la hauteur d'eau évaporée selon que le niveau de la mare est très bas ou plus élevé.

3.2 Notations

Notons h_i la hauteur de mare au jour i. Le calcul de h_i sera effectué en deux temps.

Une première hauteur h_i^{bef} sera d'abord calculée, correspondant à la hauteur de mare au jour i avant le calcul de sa décroissance (*bef* pour *before*), c'est-à-dire en ne considérant que les apports d'eau positifs rapides.

Ensuite sera calculée $\widetilde{h}_i$, la hauteur de mare au jour i après décroissance.

Il restera enfin à vérifier que la hauteur $\widetilde{h}_i$ ainsi obtenue n'est pas supérieure à la hauteur maximale de la mare h_M. On posera le cas échéant $h_i := h_M$, sinon on conservera $h_i := \widetilde{h}_i$.

Les notations suivantes représentent respectivement, pour le jour i :

- P_i la pluviométrie de ce jour ;
- $\Delta^{dir} h_i$ l'apport de hauteur d'eau par la pluie tombant directement sur la mare ;
- $\Delta^{ruiss} h_i$ l'apport de hauteur d'eau par ruissellement sur le bassin versant (pluie indirecte) ;
- $\Delta^{evap} h_i$ la perte de hauteur d'eau par évaporation ;
- $\Delta^{infil} h_i$ la perte de hauteur d'eau par infiltration de l'eau de la mare dans le sol.

Avec les notations précédentes, les hauteurs recherchées sont obtenues de la manière suivante :

$$\begin{aligned} h_i^{bef} &= h_{i-1} + \Delta^{dir} h_i + \Delta^{ruiss} h_i \\ \widetilde{h}_i &= h_i^{bef} - \Delta^{evap} h_i - \Delta^{infil} h_i \\ h_i &= \widetilde{h}_i 1_{\{0 \leqslant \widetilde{h}_i \leqslant h_M\}} + h_M 1_{\{\widetilde{h}_i > h_M\}} \end{aligned}$$

Il reste maintenant à modéliser les Δh_i.

3.3 Modélisation

3.3.1 Calcul de $\Delta^{dir} h_i$

Le plus simple et le plus évident à calculer est $\Delta^{dir} h_i$. On a directement

$$\Delta^{dir} h_i = P_i$$

3.3.2 Calcul de $\Delta^{ruiss} h_i$

Pour calculer les autres Δh_i, il est nécessaire d'introduire des paramètres relatifs à la porosité du sol :

- Notons α_1 le coefficient de porosité pour un sol sec, c'est-à-dire le pourcentage d'infiltration de l'eau sur le pourtour de la mare en une journée sur un sol sec. Ce coefficient sera vraisemblablement proche de 1.
- Notons ensuite $\alpha(i)$ le pourcentage d'infiltration sur le pourtour de la mare au jour i. Si le sol est tout à fait sec au jour i, on aura $\alpha(i) = \alpha_1$.

Il est naturel de penser que $\alpha(i)$ dépendra de α_1 et d'un facteur multiplicatif f dépendant de la quantité d'eau retenue dans le sol au jour i autour de la mare. Autrement dit ce facteur dépendra de la pluviométrie des jours précédents : $f = f(P_{i-1}, P_{i-2}, \cdots, P_{init})$. Nous l'appellerons coefficient de saturation du sol en eau.

$$\alpha(i) = \alpha_1 f(P_{i-1}, P_{i-2}, \cdots, P_{init})$$

Ce coefficient sera modélisé par une logistique.

$$f(P_{i-1}, P_{i-2}, \cdots, P_1) = \frac{\alpha_2}{\alpha_2 + \sum_{1 \leqslant j < i} P_j a_{i,j}}$$

Celle-ci traduit le fait que plus il aura plu les jours précédents, plus le sol sera saturé en eau, et moindre sera la proportion d'eau à s'infiltrer au jour i. Le choix de la modélisation par une logistique se justifie par le fait que sa forme est bien représentative des phénomènes de saturation. En effet si on passe à un temps continu et que l'on note

$$P_t = \sum_{1 \leqslant j < i} P_j a_{i,j}$$

$$f_t = f(P_{i-1}, P_{i-2}, \cdots, P_{init})$$

on a

$$\frac{f_t'}{f_t} = -\frac{(\alpha_2 + P_t)'}{\alpha_2 + P_t}$$

autrement dit la variation relative du coefficient de saturation f_t est égale à la variation relative de la porosité $\alpha_2 + P_t$.

On a :

$$\underbrace{0}_{\text{sol saturé}} \leqslant f(\cdots) \leqslant \underbrace{1}_{\text{sol sec}}$$

Naturellement l'influence de la pluviométrie d'un jour j antérieur au jour i considéré est inversement proportionnelle à l'écart $i - j$. Ainsi on pourra modéliser les coefficients $a_{i,j}$ comme des fonctions décroissantes de $i - j$ uniquement :

$$a_{i,j} = \beta^{i-j} \ , \ 0 < \beta < 1$$

Le volume d'eau qui ruissellera sur le bassin versant au jour i sera alors égal à la surface non immergée de la mare, multipliée par la pluviométrie du jour i, multipliée enfin par le pourcentage d'eau qui ne s'infiltrera pas dans le sol autour de la mare, autrement dit $1 - \alpha(i)$.

Notons :
– S_M la surface maximale du bassin versant ;
– S_{fond} la surface du fond (plat) de la mare ;
– $S(h)$ la surface d'eau de la mare correspondant à une hauteur d'eau h.

La relation surface-niveau n'étant connue pour les mares étudiées que par pas de 5 cm, nous approximerons les données manquantes grâce à la relation de Thalès.

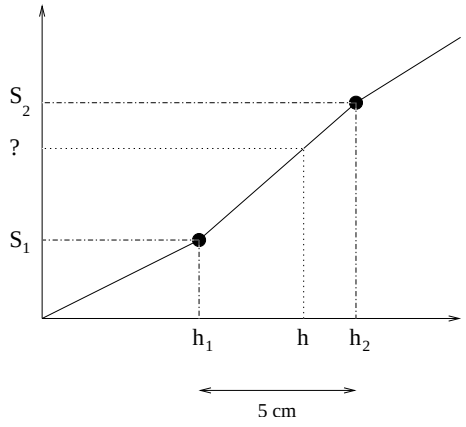


FIG. 3.2 – Approximation des données manquantes

Le volume d'eau qui ruissellera sur le bassin versant au jour i est :

$$V_{ruiss} = P_i(1 - \alpha(i))(S_M - S(h_{i-1}))$$

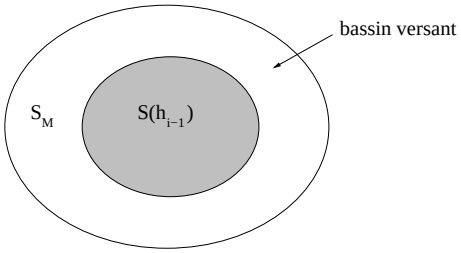


FIG. 3.3 – Surface du bassin versant

Pour le ramener à l'apport de hauteur d'eau correspondant dans la mare, on effectuera l'une des deux approximations suivantes :

La première, la plus grossière, consiste à se baser sur ce schéma :

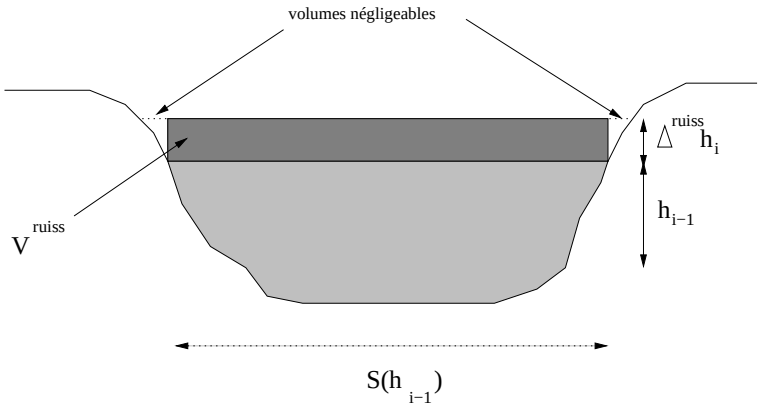


FIG. 3.4 – Approximation grossière de la hauteur

et à poser

$$\Delta^{ruiss} h_i = \frac{V_{ruiss}}{S(h_{i-1})}$$

L'autre, moins drastique, consiste à approcher le volume V_{ruiss} par des volumes rectangulaires de 5 cm de hauteur (afin d'utiliser les surfaces connues) et en déduire $\Delta^{ruiss} h_i$ (cf schéma page suivante)

Pour ce faire, posons

– H_{i-1} la hauteur immédiatemement supérieure à h_{i-1} et multiple de $\Delta = 0.05$

– $V(k) = (H_{i-1} - h_{i-1}) * S(h_{i-1}) + \sum_{l=0}^{k} \Delta * S(H_{i-1} + l * \Delta)$

On va approcher V_{ruiss} en incrémentant k. Soit

$$k_0 = \sup\{k : V(k) \leqslant V_{ruiss}\}$$

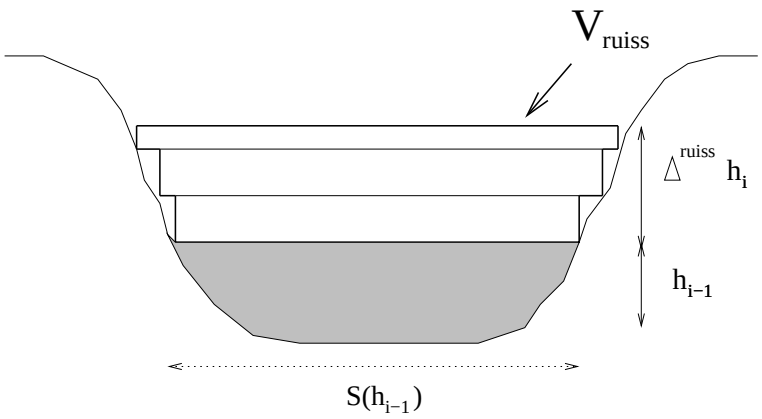


FIG. 3.5 – Approximation plus fine de la hauteur

Alors on aura

$$V_{ruiss} = V(k_0) + [\Delta^{ruiss} h_i + h_{i-1} - (H_{i-1} + 0.05 * k_0)] * S(H_{i-1} + 0.05 * k_0)$$

D'où finalement

$$\Delta^{ruiss} h_i = \frac{V_{ruiss} - V(k_0)}{S(H_{i-1} + 0.05 * k_0)} + H_{i-1} + 0.05 * k_0 - h_{i-1}$$

Les calculs ici ne sont pas exhaustifs, car tous les cas ne sont pas pris en compte. Par exemple, si h_{i-1} est nulle, on ne travaillera pas avec $S(h_{i-1})$ (qui sera nulle également) mais avec la surface du fond S_{fond}. Pour plus de détails, voir **Programmation du modèle**.

Nous verrons plus tard (dans le chapitre **Premiers ajustements**) pourquoi cette deuxième approximation plus fine est plus adaptée.

3.3.3 Calcul de $\Delta^{evap} h_i$

On modélise $\Delta^{evap} h_i$ de la façon suivante :

$$\Delta^{evap} h_i = E_1 1_{\{0 < h_i^{bef} \leqslant h_E\}} + E_2 1_{\{h_i^{bef} > h_E\}}$$

où h_E est un paramètre représentant un seuil de hauteur d'eau en-dessous duquel on considère que la hauteur évaporée en une journée est de E_1, et au-dessus duquel elle sera de E_2.

L'évaporation étant plus intense lorsque la mare est presque à sec, on suppose $E_1 \geqslant E_2$.

3.3.4 Calcul de $\Delta^{infil}h_i$

La modélisation de $\Delta^{infil}h_i$ suit le même principe que celle de $\Delta^{ruiss}h_i$: en effet la hauteur d'eau par infiltration va être obtenue en multipliant h_i^{bef} par un coefficient $\widetilde{\alpha}(i)$ traduisant le pourcentage d'eau infiltrée sous la mare.

Ce coefficient $\widetilde{\alpha}(i)$ sera similaire au $\alpha(i)$ calculé précédemment, à ceci près que l'influence d'un jour j antérieur ne proviendra non plus de la pluviométrie P_j mais de la hauteur d'eau de la mare h_j en ce jour - sauf si celle-ci est nulle, auquel cas on considérera P_j.

D'où la modélisation de $\Delta^{infil}h_i$:

$$\Delta^{infil}h = h_i^{bef}\widetilde{\alpha}(i)$$

où

$$\widetilde{\alpha}(i) = \alpha_1 f(P_{i-1}1_{\{h_{i-1}=0\}} + h_{i-1}, \cdots)$$

soit

$$\widetilde{\alpha}(i) = \alpha_1 \frac{\alpha_2}{\alpha_2 + \sum_{1 \leqslant j < i}(P_j 1_{\{h_j=0\}} + h_j)a_{i,j}}$$

3.4 Résumé du modèle

Rassemblons ici les formules utilisées dans le modèle :

3.4.1 Calcul de h_i^{bef}

$$h_i^{bef} = h_{i-1} + \Delta^{dir} h_i + \Delta^{ruiss} h_i$$

avec

$$\Delta^{dir} h_i = P_i$$

$$\Delta^{ruiss} h_i = \frac{V_{ruiss} - V(k_0)}{S(H_{i-1} + 0.05 * k_0)} + H_{i-1} + 0.05 * k_0 - h_{i-1}$$

où

$$\begin{aligned}
V_{ruiss} &= P_i(1-\alpha(i))(S_M - S(h_{i-1})) \\
\alpha(i) &= \alpha_1 \frac{\alpha_2}{\alpha_2 + \sum_{1 \leqslant j < i} P_j a_{i,j}} \\
a_{i,j} &= \beta^{i-j} \ , \ 0 < \beta < 1 \\
H_{i-1} &= \text{hauteur immédiatemement supérieure} \\
&\quad \text{à } h_{i-1} \text{ et multiple de } 0.05 \\
V(k) &= (H_{i-1} - h_{i-1}) * S(h_{i-1}) + \sum_{l=0}^{k} \Delta * S(H_{i-1} + l * \Delta) \\
k_0 &= \sup\{k : V(k) \leqslant V_{ruiss}\}
\end{aligned}$$

3.4.2 Calcul de $\widetilde{h_i}$

$$\widetilde{h_i} = h_i^{bef} - \Delta^{evap} h_i - \Delta^{infil} h_i$$

avec

$$\Delta^{evap} h_i = E_1 1_{\{0 < h_i^{bef} \leqslant h_E\}} + E_2 1_{\{h_i^{bef} > h_E\}}$$

$$\Delta^{infil} h = h_i^{bef} \widetilde{\alpha}(i)$$

où

$$\widetilde{\alpha}(i) = \alpha_1 \frac{\alpha_2}{\alpha_2 + \sum_{1 \leqslant j < i} (P_j 1_{\{h_j = 0\}} + h_j) a_{i,j}}$$

3.4.3 Calcul de h_i

$$h_i = \widetilde{h}_i 1_{\{0 \leqslant \widetilde{h}_i \leqslant h_M\}} + h_M 1_{\{\widetilde{h}_i > h_M\}}$$

3.5 Paramètres

Le modèle fera ainsi intervenir plusieurs paramètres, dont certains sont des paramètres physiques qui nous sont directement accessibles grâce aux données. C'est le cas de :

- la hauteur maximale h_M
- la surface maximale S_M
- la surface de fond S_{fond}

Il reste ainsi 6 paramètres à estimer :

- le "seuil d'évaporation" h_E
- les hauteurs E_1 et E_2 représentant les pertes de hauteur d'eau par évaporation (selon que l'on est au-dessus du seuil h_E ou non)
- le coefficient α_1 correspondant au coefficient d'infiltration en une journée sur un sol sec ($0 < \alpha_1 < 1$, vraisemblablement proche de 1)
- le coefficient α_2 qui apparait dans la logistique intervenant dans le cacul de $\Delta^{ruiss} h_i$ et de $\Delta^{infil} h_i$
- le coefficient β qui traduit la "mémoire de saturation" du modèle, autrement dit l'influence des jours précédents (cf calcul de $\Delta^{ruiss} h$). On suppose $0 < \beta < 1$.

Nous verrons plus tard qu'il s'avérera judicieux de fixer les paramètres h_E, E_1 et E_2 (pour toutes les mares) et de restreindre l'estimation aux 3 paramètres α_1, α_2 et β.

Chapitre 4

Premiers ajustements

4.1 Approximation trop grossière de $\Delta^{ruiss} h_i$

Pour tester la cohérence de notre modèle, nous avons travaillé sur l'évolution de la mare de Furdu durant l'année 2003. Nous avions en notre possession les données de pluviométrie de cette année au niveau de la mare, ainsi que les hauteurs observées de cette mare. Malheureusement, ni les relevés limnimétriques ni les relevés pluviométriques n'ayant commencé dès la première pluie, nous avons dû compléter les quelques jours manquants par ce qui d'après les années précédentes nous paraissait vraisemblable. Il reste cependant près de deux semaines (du 31 août au 12 septembre) où nous ne possédons aucune donnée limnimétrique.

Nous avons pu observer grâce aux données limnimétriques des années 2001, 2002 et 2003 que, en l'absence de pluie (l'évaporation est alors l'unique cause de décroissance) et en fin de saison (saturation du sol maximum), la mare perdait en moyenne 2 cm d'eau par jour lorsque son niveau était supérieur à un certain seuil situé aux alentours de 25 cm, et qu'en-dessous de ce seuil elle perdait environ 1 cm d'eau par jour.

Ceci nous a permis de prendre des paramètres raisonnables ($h_E = 0.25$, $E_1 = 0.02$, $E_2 = 0.01$) pour lancer les premières simulations. Nous avons choisi un coefficient de porosité pour un sol sec relativement proche de 1 : $\alpha_1 = 0.9$. Enfin nous avons pris les valeurs $\alpha_2 = 0.01$ et $\beta = 0.7$.

Voici ce que nous avons obtenu lors de cette première simulation :

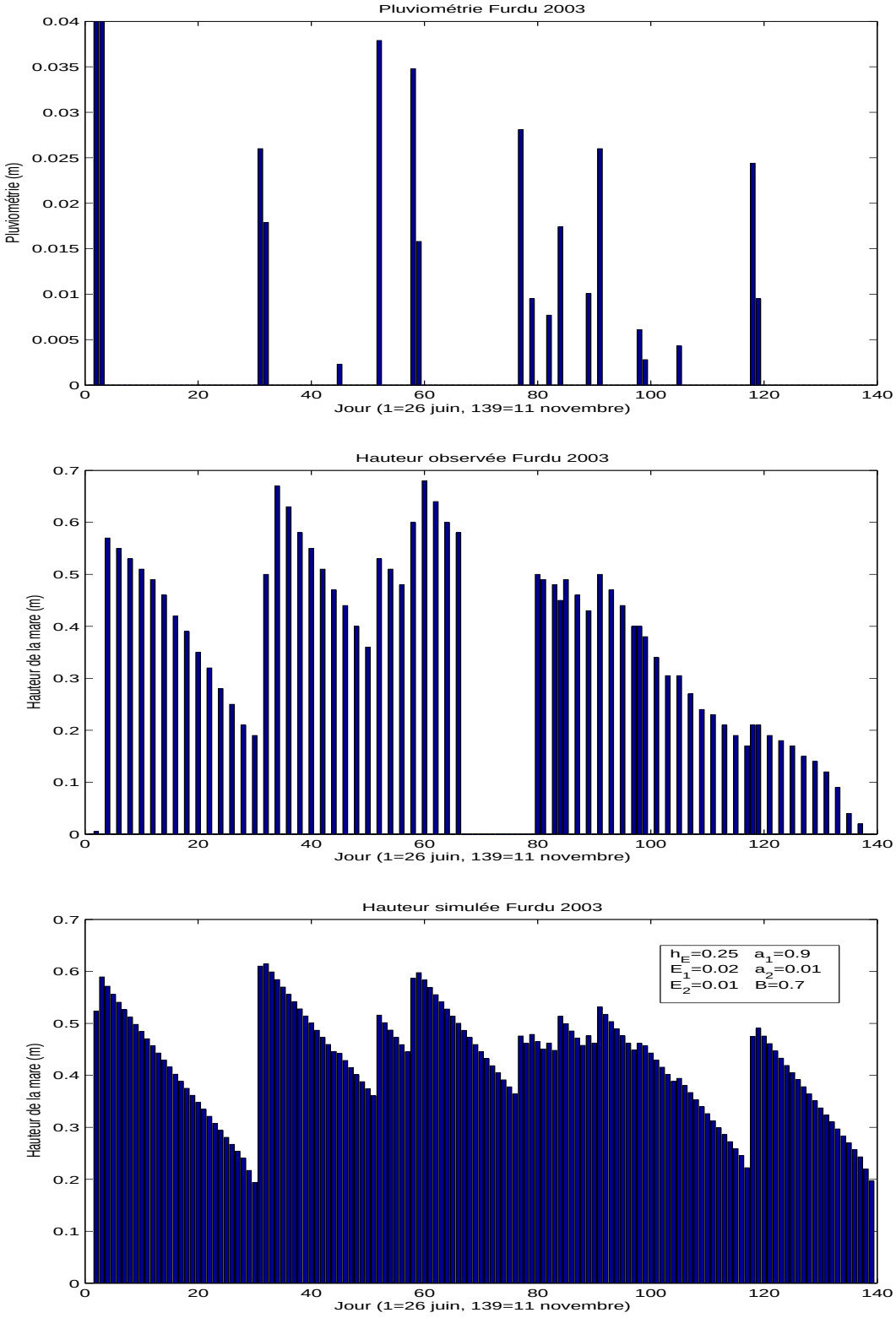


FIG. 4.1 – Première simulation

Ce premier résultat semblait satisfaisant. Nous avons cependant testé ce modèle sur des petites hauteurs, afin de connaître sa réaction lors d'un remplissage brutal de la mare. Pour cela, il a suffi d'augmenter artificiellement la décroissance, en faisant par exemple passer E_2 de 0.01 à 0.015. On a ainsi accru la décroissance par évaporation, ce qui a amené le programme à atteindre de plus petites hauteurs que précédemment.

Si l'on augmente la décroissance, alors on diminue la saturation du sol en eau, et donc on devrait *diminuer* la hauteur des pics. Or, voici le résultat de cette deuxième simulation :

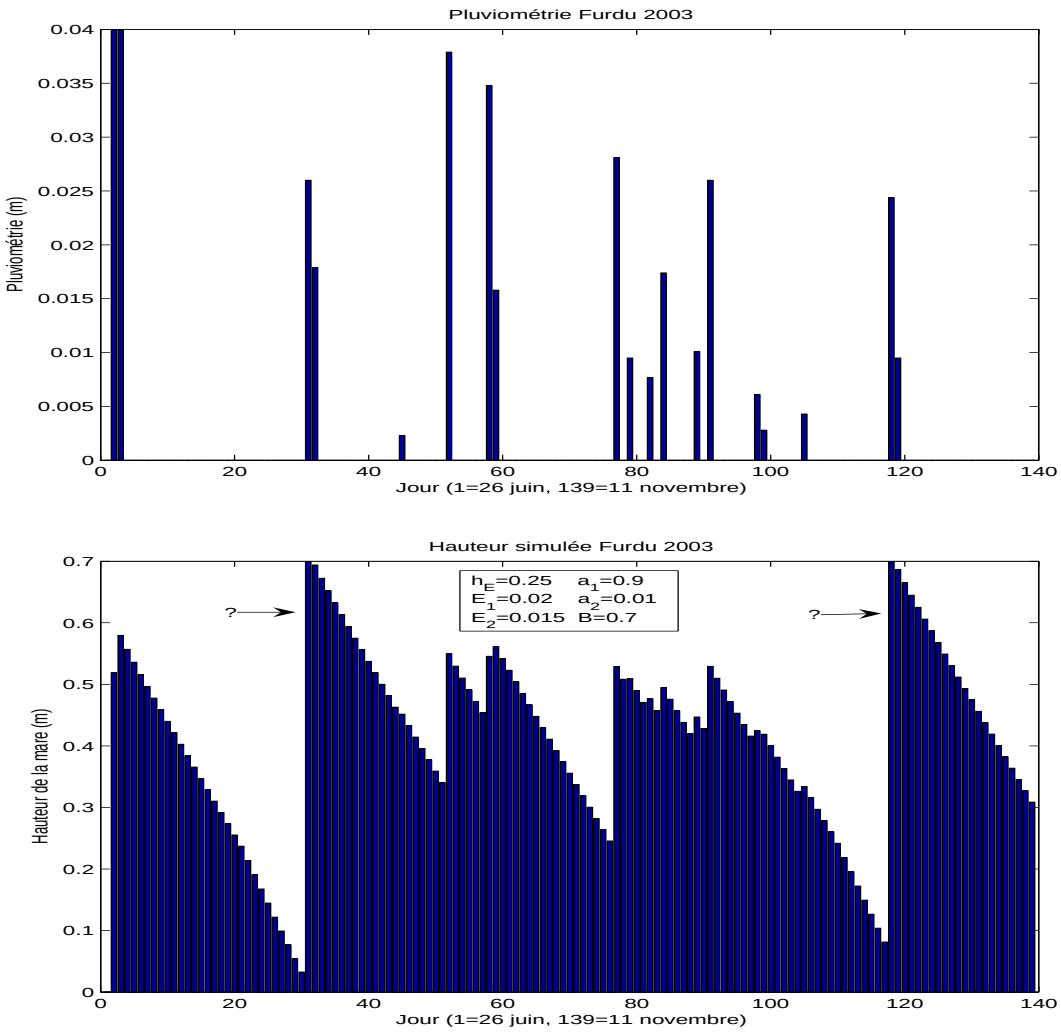


FIG. 4.2 – Test sur les petites hauteurs

La réaction du modèle aux petites hauteurs n'est donc pas satisfaisante, puisqu'on observe au contraire une croissance des pics (marqués d'une flèche).

Cela est dû à l'approximation grossière de $\Delta^{ruiss}h_i$ qui, dans le cas où le niveau de la mare est bas et l'apport d'eau important, donne une valeur trop élevée.

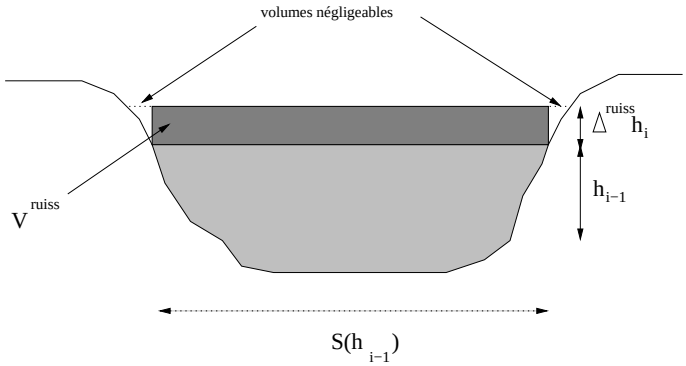


FIG. 4.3 – Approximation grossière de la hauteur

En effet les volumes près du bord qui n'étaient pas pris en compte dans le calcul ne sont dans ce cas plus négligeables, et la surface d'eau $S(h_{i-1})$ étant très faible, la hauteur d'eau $\Delta^{ruiss}h_i$ ainsi obtenue était très importante.

C'est pour cette raison que nous avons privilégié l'approximation plus fine de $\Delta^{ruiss}h_i$ (présentée dans le paragraphe **Calcul de** $\Delta^{ruiss}h_i$), qui nous a fourni des résultats plus cohérents sur les petites hauteurs.

4.2 Surface trop faible du bassin versant

L'examen de plusieurs simulations nous a également permis de remarquer que les croissances causées par des chutes de pluie étaient généralement moins prononcées que ce qu'on pouvait observer dans la réalité. Or la croissance du niveau de la mare (de plusieurs dizaines de centimètres) étant essentiellement due au phénomène de ruissellement (la pluie directe n'ajoutant que quelques centimètres), nous en avons conclu que la modélisation du ruissellement n'était pas encore satisfaisante.

Cela était en fait tout simplement dû à une mauvaise estimation de la surface du bassin versant. Cette surface était tout d'abord assimilée à la surface maximale observée de la mare, autrement dit 17 030 m^2 pour la mare de Furdu. Cette dernière se trouve sur un "bouclier", c'est-à-dire un sol assez plat, et le ruissellement vers la mare est observé sur une surface bien plus étendue que celle correspondant au bassin même. N'ayant pas de modèle numérique de terrain pour cette zone nous n'avons pas pu avoir d'estimation de cette surface, mais il est vraisemblable que celle-ci soit de l'ordre de plusieurs hectares. Nous avons ainsi corrigé la valeur de S_M en la fixant

à 30 000 m^2.

Les deux graphiques suivants représentent deux simulations effectuées avec les mêmes paramètres, hormis la surface du bassin versant (17 030 m^2 puis 30 000 m^2).

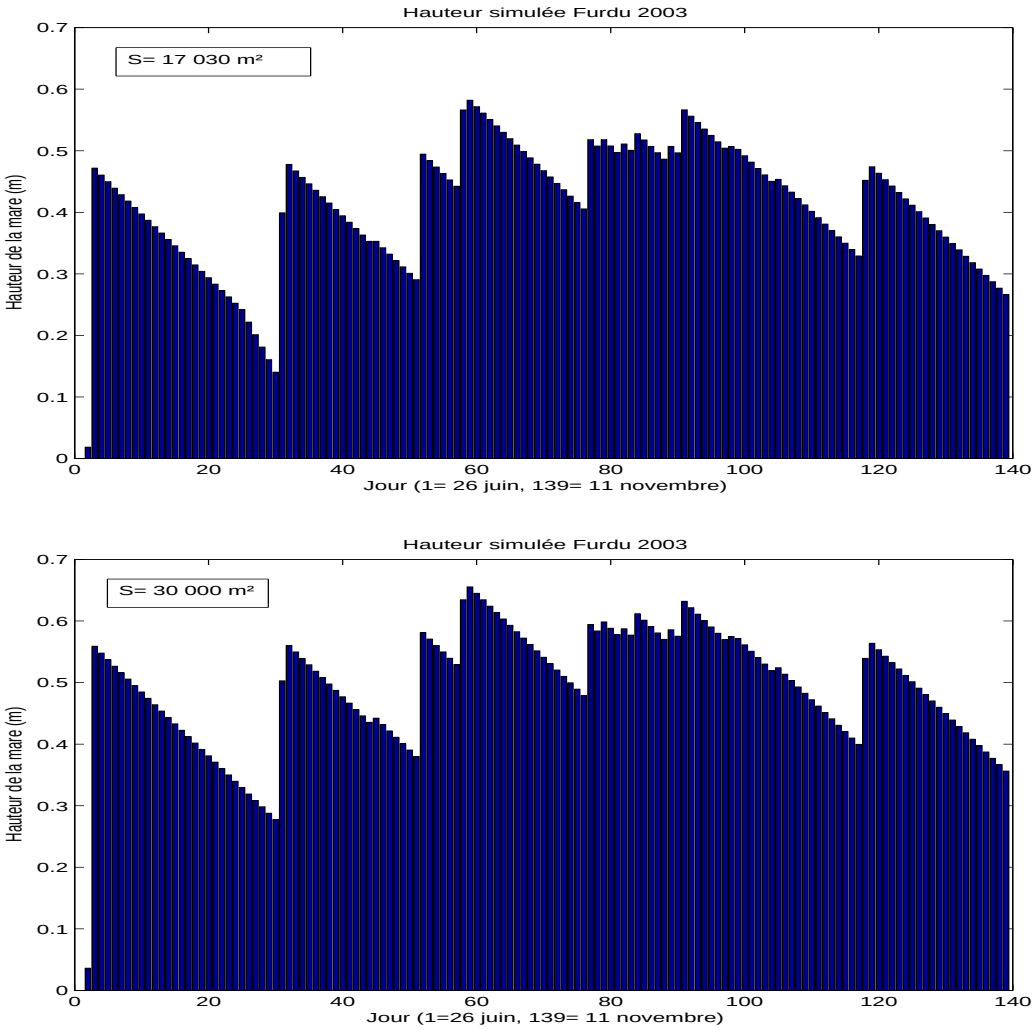


FIG. 4.4 – Simulation avec deux surfaces différentes pour le bassin versant ($h_E = 0.25,\ E_1 = 0.02,\ E_2 = 0.01,\ a_1 = 0.9,\ a_2 = 0.001,\ B = 0.7$)

Chapitre 5

Analyse de sensibilité

L'influence des paramètres relatifs à la décroissance par évaporation est facilement visible :

- si l'on modifie h_E, on change le seuil à partir duquel la décroissance par évaporation va être différente ;
- si l'on modifie E_1 ou E_2, on modifie de manière significative la vitesse de décroissance, notamment durant les périodes où il ne pleut pas, puisqu'alors l'évaporation est l'unique cause de décroissance.

Nous nous sommes donc spécialement intéressées à l'influence - plus complexe - des paramètres α_1, α_2 et β.

D'après le modèle, on sait que si l'on augmente α_1 - c'est-à-dire la porosité du sol sec - tous les pics vont diminuer, avec un effet plus marqué sur le premier pic au début de la saison des pluies. C'est ce qu'on retrouve de manière quantitative sur les deux graphiques suivants, qui comparent deux simulations pour $\alpha_1 = 0.9$ et $\alpha_1 = 0.1$.

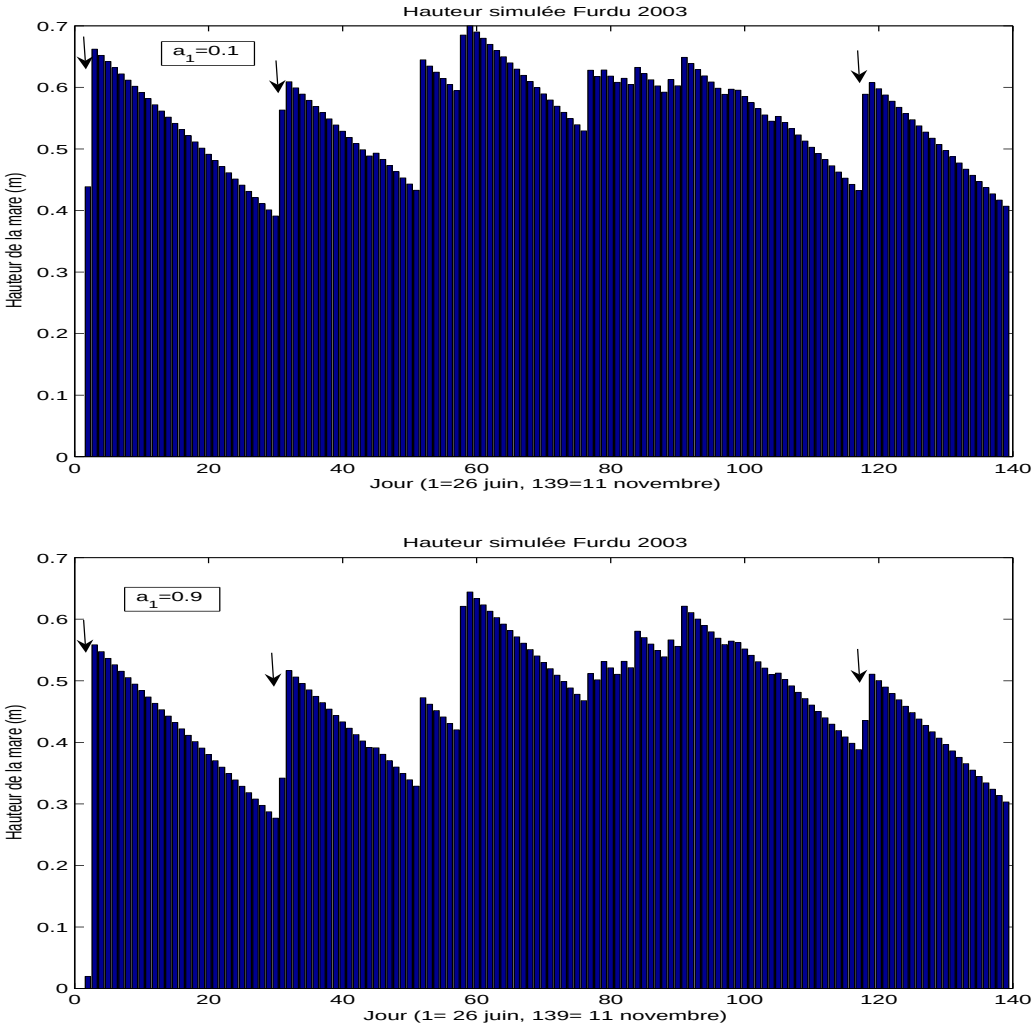


FIG. 5.1 – Simulation pour deux valeurs de α_1 ($h_E = 0.25, E_1 = 0.02, E_2 = 0.01, a_2 = 0.001, B = 0.7$)

Analysons maintenant la sensibilité du modèle aux paramètres α_2 et β, qui interviennent tous les deux dans les calculs de $\Delta^{ruiss}h_i$ et de $\Delta^{infil}h_i$; ou plus exactement dans les calculs de $\alpha(i)$ (resp. $\widetilde{\alpha}(i)$), pourcentage d'infiltration sur le pourtour de la mare (resp. sous la mare) à chaque jour i.

$$\alpha(i) = \alpha_1 \frac{\alpha_2}{\alpha_2 + \sum_{1 \leqslant j < i} P_j a_{i,j}}$$

$$a_{i,j} = \beta^{i-j} \ , \ 0 < \beta < 1$$

On peut déjà dire qu'augmenter α_2 va augmenter $\alpha(i)$ et donc diminuer la saturation : tous les pics vont décroître. De plus si α_2 est très grand, $\alpha_2 \gg \sum P_j a_{i,j}$ et donc $\alpha(i) \simeq \alpha_1$, c'est-à-dire le pourcentage d'infiltration maximum.

Quant à β, son augmentation va accroître la capacité de mémoire du modèle, c'est-à-dire augmenter la saturation.

Les graphiques suivants vont permettre une illustration quantitative de l'influence de ces deux paramètres. Ils représentent des simulations qui ont été effectuées sur la mare de Furdu avec les données pluviométriques de l'année 2003, avec les paramètres suivants :

$$\begin{aligned} h_E &= 0.25 \\ E_1 &= 0.02 \\ E_2 &= 0.01 \\ \alpha_1 &= 0.9 \end{aligned}$$

On peut constater au vu de ces graphiques que l'influence de β est surtout visible sur la fin de la saison des pluies : la croissance de β augmente la hauteur de mare relative en fin de saison par rapport au début de saison.

On peut également remarquer que plus α_2 est petit, moins le modèle est sensible aux variations de β. De même si β est suffisamment grand, le modèle est moins sensible aux variations de α_2.

Notons enfin que α_1, α_2 et β jouent un rôle dans la forme de la décroissance des pics de hauteur :

- Si α_1 et α_2 sont grands le pic devrait commencer par décroître rapidement, car alors l'infiltration est plus importante que l'évaporation. Puis, la saturation du sol en eau augmentant, l'évaporation devient plus importante que l'infiltration et la décroissance se termine de façon linéaire. C'est ce qu'on peut par exemple observer sur l'un des graphiques suivants pour $\alpha_2 = 0.1$, $\beta = 0.9$, tout particulièrement autour des jours n°60 et 120.
- Lorsque α_2 diminue alors l'infiltration devient négligeable par rapport à l'évaporation, et on a une décroissance linéaire, ce qu'on observe par exemple pour le cas où $\alpha_2 = 0.001$.

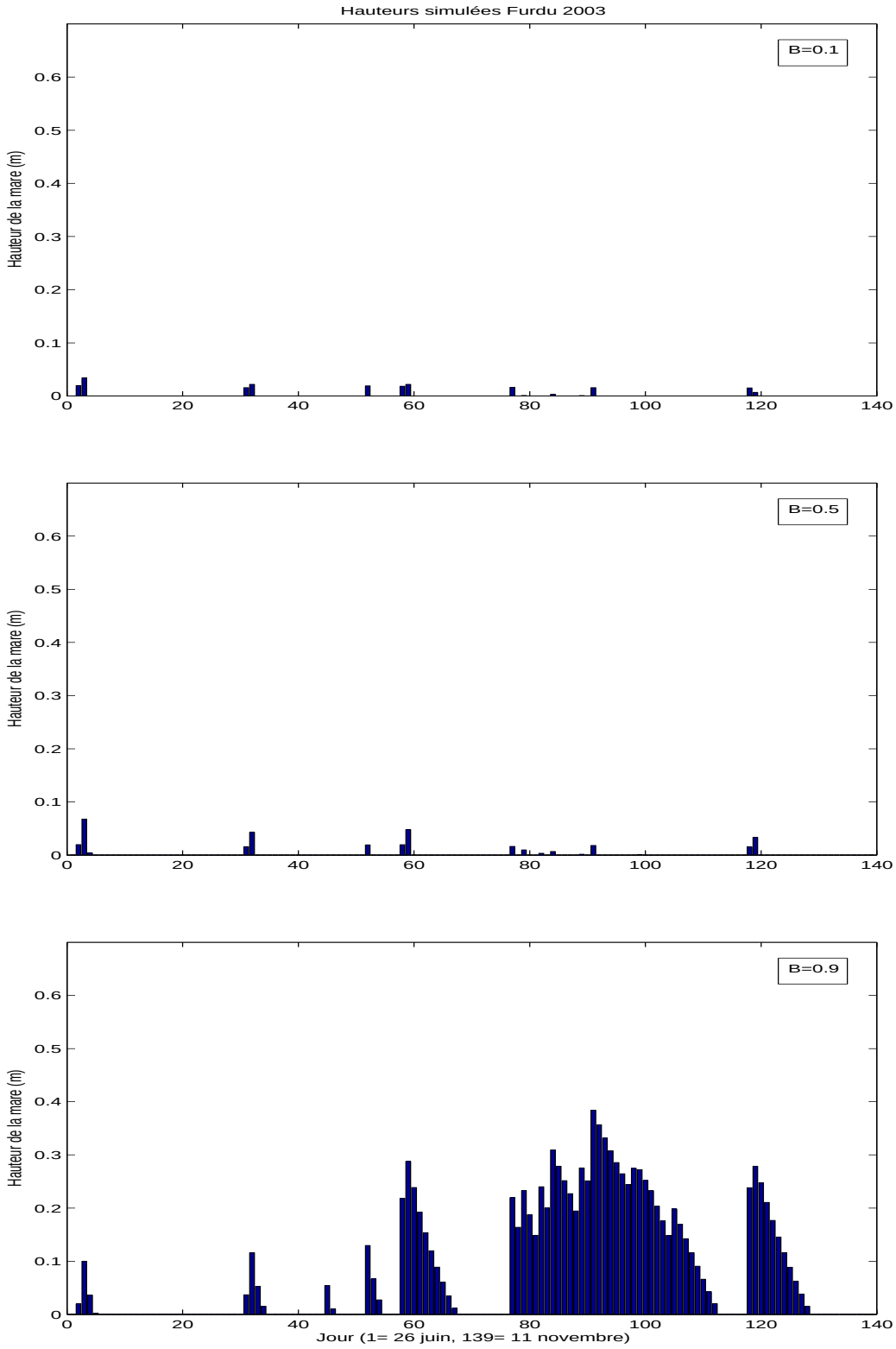


FIG. 5.2 – Influence de β pour $\alpha_2 = 0.1$

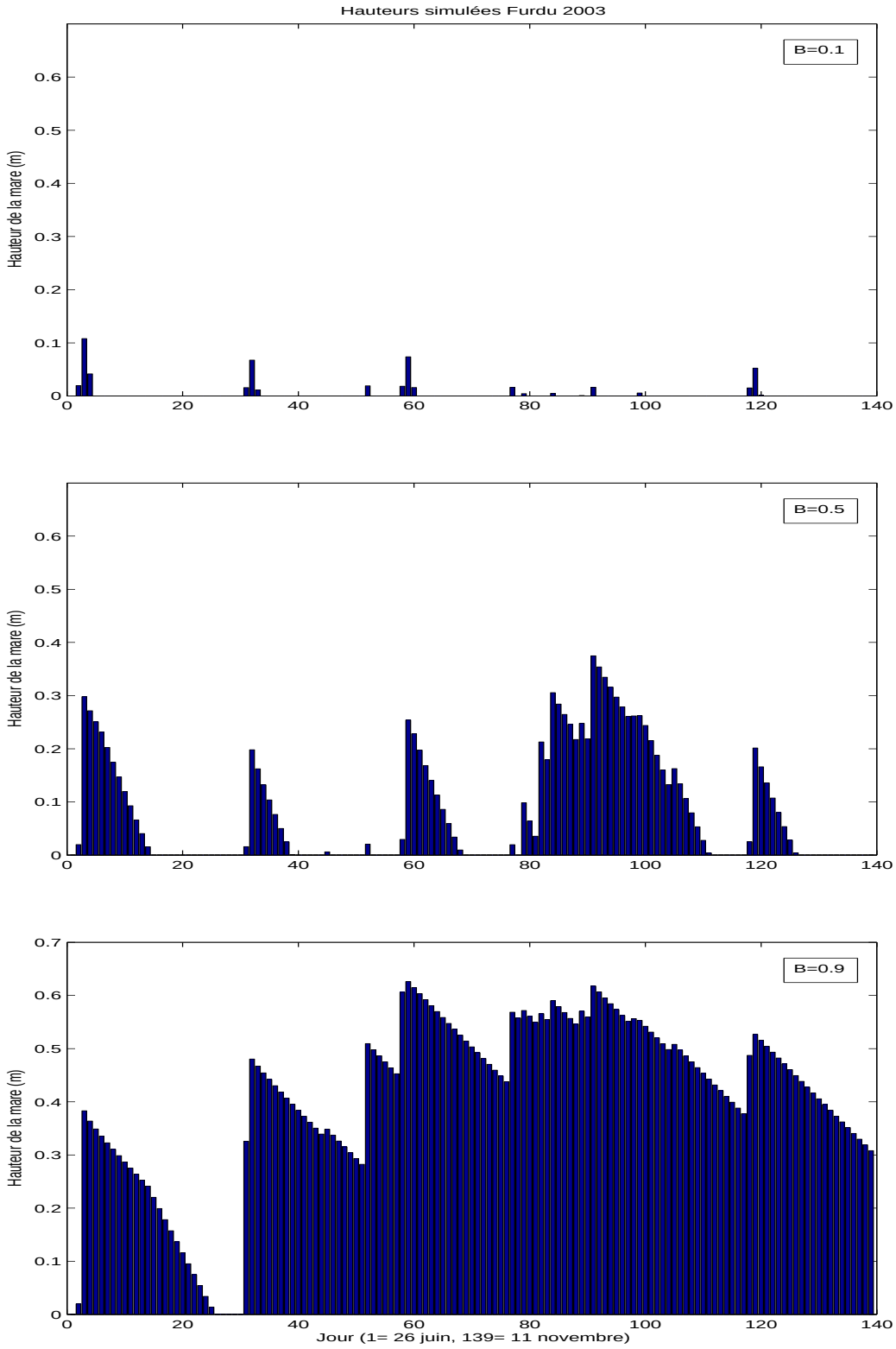


FIG. 5.3 – Influence de β pour $\alpha_2 = 0.01$

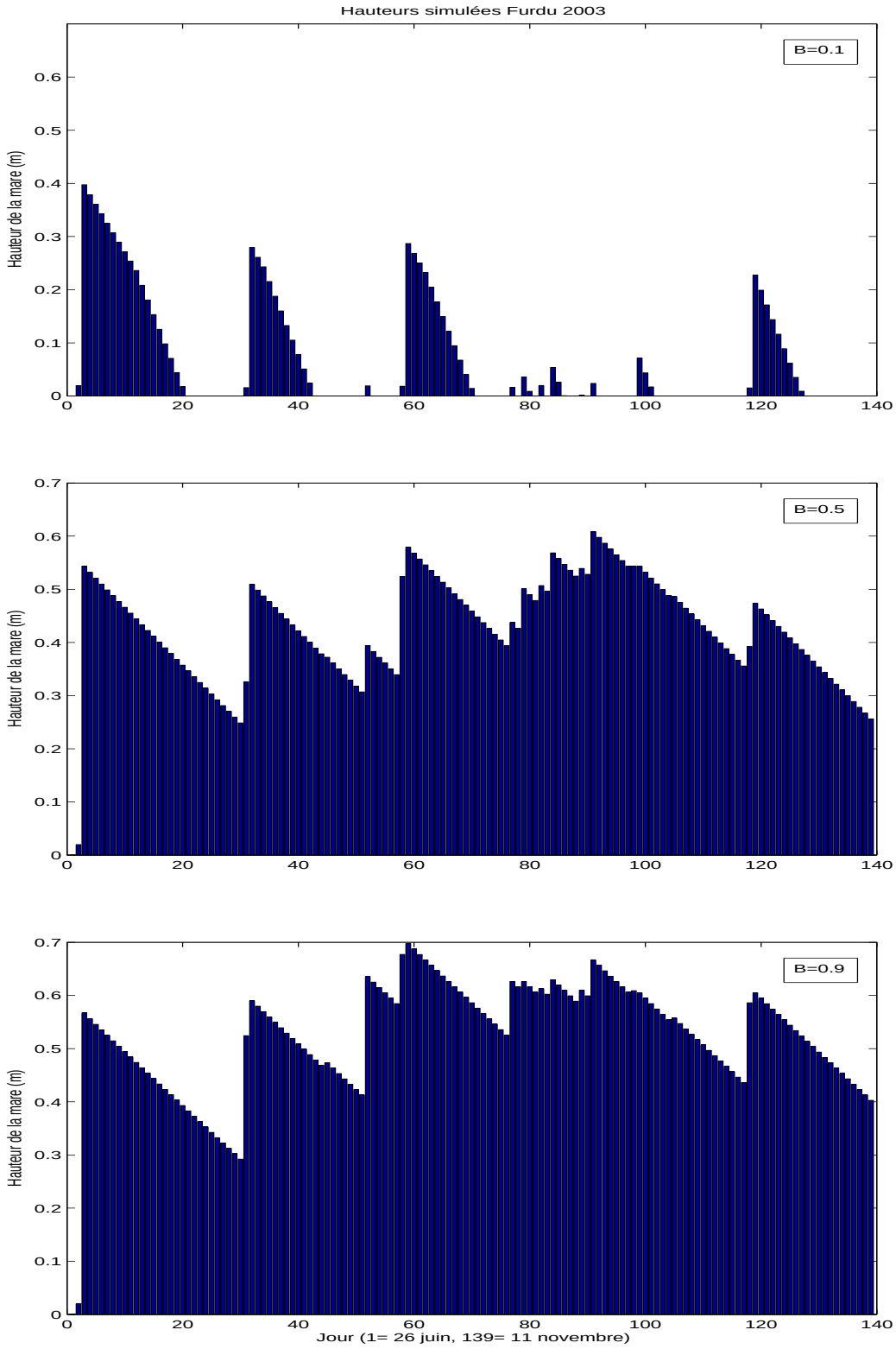


FIG. 5.4 – Influence de β pour $\alpha_2 = 0.001$

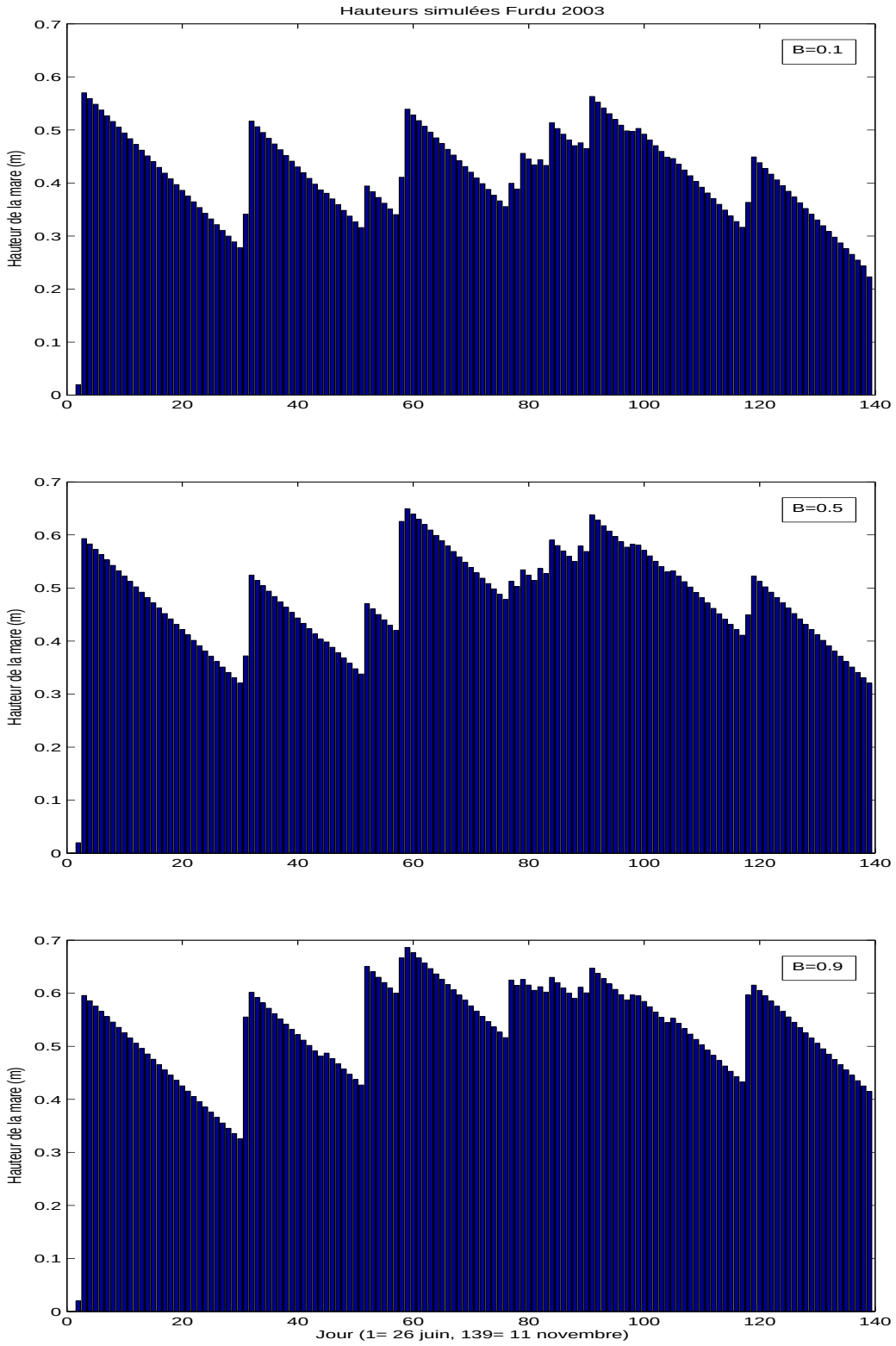


FIG. 5.5 – Influence de β pour $\alpha_2 = 0.0001$

Chapitre 6

Estimation des paramètres

6.1 Restriction du nombre de paramètres à estimer

Nous allons estimer les paramètres par la méthode des moindres carrés ordinaires :

$$\widehat{\theta}_n = \text{argmin}_\theta \sum_{i=1}^{n} (h^{obs}(i) - h_\theta(i))^2$$

où $h^{obs}(i)$ est la hauteur observée au jour i, et $h_\theta(i)$ la hauteur modélisée au jour i, avec le paramètre

$$\theta = (h_E, E_1, E_2, \alpha_1, \alpha_2, \beta)$$

L'entier n est l'indice du dernier jour d'observation (en 2003, $n = 139$).

Avant de travailler avec des observations réelles, nous avons testé la méthode d'estimation sur des simulations, en posant

$$h^{obs}(i) = h_{\theta_0}(i)$$

pour un certain paramètre θ_0 qui produit une simulation plausible :

$$\theta_0 = (.25, .02, .01, .9, .0001, .7)$$

Nous avons utilisé la fonction *lsqnonlin* de Matlab qui, en partant d'une valeur initiale θ_1, recherche le θ qui minimise $\sum_i (h_{\theta_0}(i) - h_\theta(i))^2$.

L'idéal aurait été d'obtenir comme réponse quelque chose de proche de θ_0, et ce quelle que soit la valeur initiale θ_1.

Or ce n'est pas le cas :

– Si l'on prend comme valeur initiale

$$\theta_1 = (.3, .03, .02, .8, .0002, .6)$$

la fonction nous retourne

$$\widehat{\theta}_n = (.3, .0458, .0101, .8822, .0001, .6848)$$

– Si l'on prend comme valeur initiale

$$\theta_1 = (.1, .06, .05, .7, .001, .8)$$

la fonction nous retourne

$$\widehat{\theta}_n = (.1, .06, .05, .7, .001, .8)$$

Les réponses sont donc trop éloignées du θ_0 que l'on aurait souhaité obtenir. Cet écart est dû à un problème d'identifiabilité de certains paramètres, et d'après les réponses obtenues il semble que le problème vienne surtout des trois premiers paramètres (h_E, E_1, E_2). Les valeurs obtenues pour $(\alpha_1, \alpha_2, \beta)$ sont par ailleurs plutôt proches de celles de θ_0 correspondantes.

Ceci nous amène à restreindre l'estimation à celle des paramètres $(\alpha_1, \alpha_2, \beta)$, et fixer les 3 restants grâce aux valeurs que nous avions pu déterminer en observant les données réelles :

$$\begin{aligned} h_E &= 0.25 \\ E_1 &= 0.02 \\ E_2 &= 0.01 \end{aligned}$$

6.2 Test de la méthode d'estimation sur des simulations

Considérons la fonction

$$S_n(\theta) = \sum_{i=1}^{n} (h_{\theta_0}(i) - h_\theta(i))^2$$

pour le vecteur de paramètres

$$\theta = (\alpha_1, \alpha_2, \beta)$$

et celui servant à la simulation

$$\theta_0 = (.9, .001, .7)$$

pour l'année 2003 (on a donc $n = 139$).

6.2.1 Visualisation graphique

Les graphiques suivants permettent de connaître l'évolution de $S_n(\theta)$ en fixant un paramètre et en faisant varier les deux autres, afin de déceler d'éventuels problèmes d'identifiabilité.

Chaque page correspond à un paramètre fixé et présente une vue large et une vue centrée autour du minimum.

Les premières vues (plus larges) amènent à penser que l'argument minimum de la fonction $S_n(\theta)$ sera difficile à déceler, car les surfaces semblent très plates . Mais un zoom autour de sa valeur (connue car égale au paramètre θ_0 produisant la simulation) permet de voir que ce n'est pas le cas.

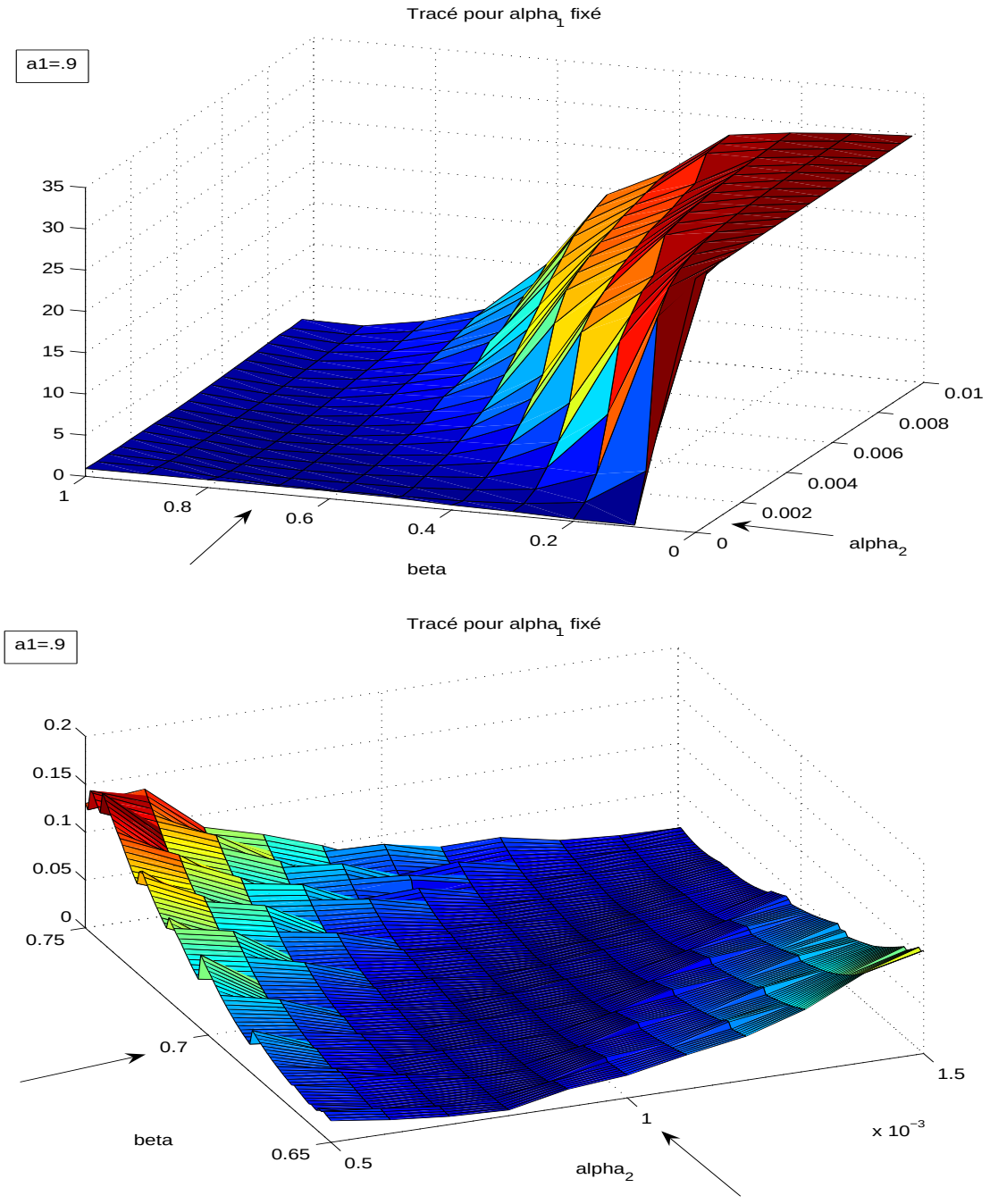


FIG. 6.1 – Evolution de $S_n(\theta)$ pour α_1 fixé

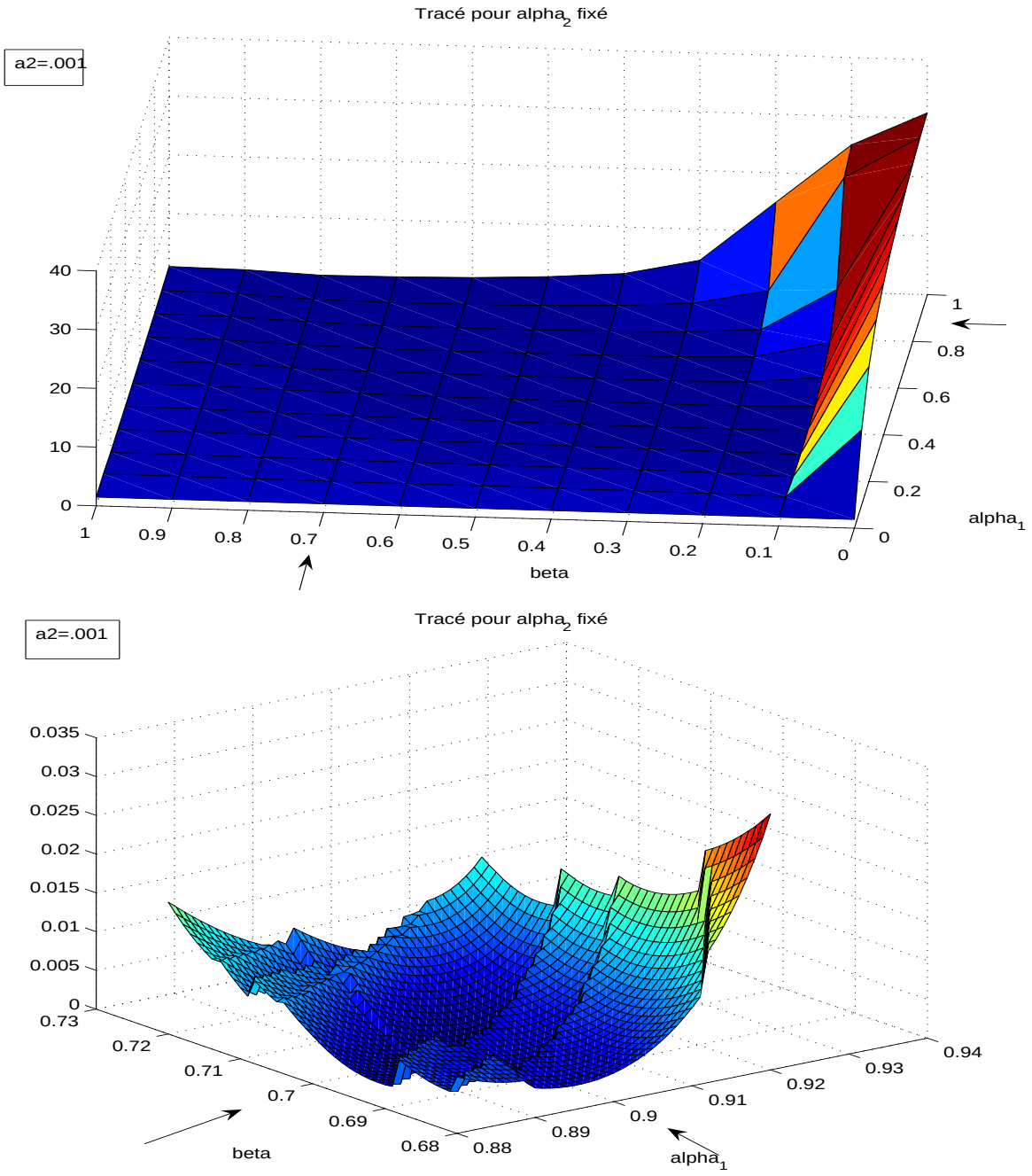


FIG. 6.2 – Evolution de $S_n(\theta)$ pour α_2 fixé

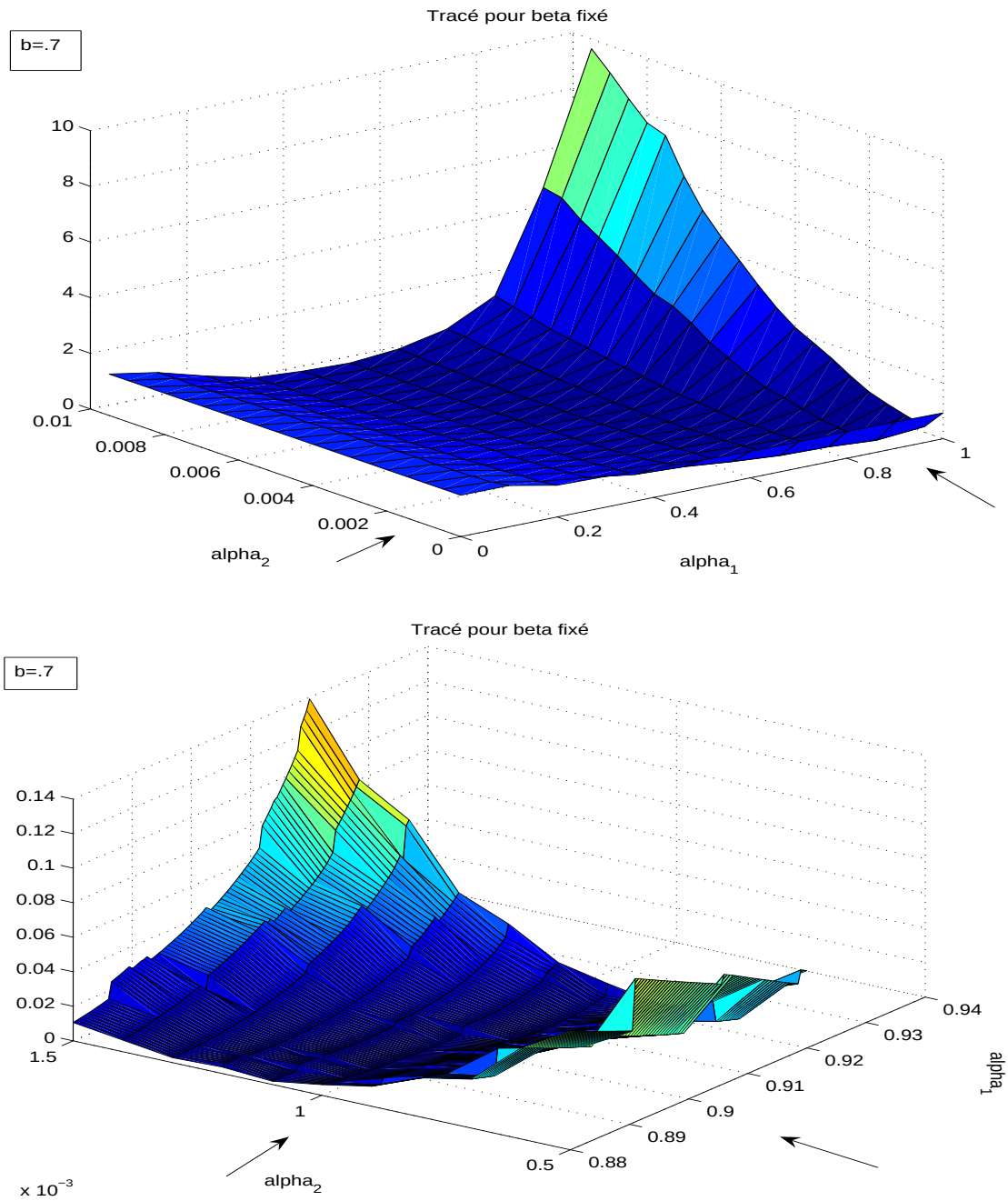


FIG. 6.3 – Evolution de $S_n(\theta)$ pour β fixé

6.2.2 Comparaison de deux méthodes

Au cours de l'étude, nous avons été amenées à constater qu'une autre méthode d'estimation fournissait des résultats plus satisfaisants que ceux obtenus par la méthode des moindres carrés ordinaires.

Elle consiste à calculer :

$$\widehat{\theta}_n = \text{argmin}_\theta \sum_{i=1}^{n} (h^{obs}(i) - h_\theta(i))^4$$

au lieu de :

$$\widehat{\theta}_n = \text{argmin}_\theta \sum_{i=1}^{n} (h^{obs}(i) - h_\theta(i))^2$$

Voici la comparaison des résultats fournis par la méthode des moindres carrés ordinaires (M_0) et cette autre méthode (M_1), pour l'année 2003.

La colonne de gauche représente le vecteur de paramètres θ_1 donné en entrée à la fonction *lsqnonlin*.

Rappelons que l'argument minimum que cette fonction devrait rechercher, et que nous connaissons, est

$$\theta_0 = (.9, .001, .7)$$

θ_1	$\widehat{\theta}_n$ obtenu par M_0	$\widehat{\theta}_n$ obtenu par M_1
(.8 .005 .6)	(.700 .003 .678)	(.836 .002 .747)
(.6 .01 .4)	(.880 .001 .71)	(.865 .001 .709)
(.1 .1 .1)	(.082 .048 .445)	(.878 .001 .682)
(.2 .2 .2)	(.881 .0012 .702)	(.840 .0008 .583)
(.3 .3 .3)	(.283 .034 .84)	(.836 .002 .747)
(1 1 1)	(.159 .157 .897)	(.879 .001 .709)

On constate que pour ces 6 valeurs de θ_1 la méthode M_1 fournit des résultats proches de la valeur θ_0 qu'on voudrait trouver, alors que la méthode des moindres carrés ordinaires ne le fait ici que pour 3 valeurs.

Cependant cette même comparaison effectuée sur les années précédentes ne fournit pas des résultats aussi probants : la méthode M_1 ne s'avère pas vraiment plus efficace que la méthode classique. Nous allons donc conserver pour la suite la méthode des moindres carrés ordinaires.

6.3 Estimation des paramètres : deux étapes

Les résultats présentés dans ce tableau nous mettent en garde quant à l'utilisation de la fonction *lsqnonlin* : si la valeur initiale θ_1 est trop éloignée du résultat θ_0, la valeur retournée risque elle aussi de l'être. C'est le cas par exemple lorsque $\theta_1 = (.1.1.1)$ (ligne 3) ou $\theta_1 = (1, 1, 1)$ (ligne 6).

C'est pourquoi une étape préliminaire à l'utilisation de la fonction *lsqnonlin* apparaît nécessaire, afin de donner en entrée une valeur θ_1 qui ne soit pas trop éloignée de l'argument minimum.

Cette étape consiste à calculer la valeur de $S_n(\theta)$ en tout point $\theta = (\alpha_1, \alpha_2, \beta)$ appartenant à un ensemble discret, en fait une "grille" à 3 dimensions, où l'on détermine l'intervalle et le pas pour α_1, α_2 et β.

La comparaison de ces valeurs nous donne alors le minimum sur cette "grille", et le paramètre θ correspondant. On peut ensuite réitérer ce processus en centrant la grille autour du θ obtenu en en diminuant les pas, autant de fois que nécessaire.

La valeur obtenue au terme de cette étape pourra ensuite être utilisée comme valeur d'entrée pour la fonction *lsqnonlin*, qui nous fournira alors le $\widehat{\theta}_n$ cherché.

6.4 Estimation des paramètres sur des observations réelles

Après s'être assuré du bon fonctionnement de cette méthode d'estimation des paramètres sur des simulations, nous pouvons l'appliquer à des observations réelles, par exemple les observations à la mare de Furdu pour l'année 2003.

Utilisons une première fois la fonction *grille* pour approcher l'argument minimum :

```
>> grille
quelle annee? 2003
simulations (1) ou observations reelles (2)? 2
valeur initiale pour alpha_1? .5
pas pour alpha_1? .05
valeur finale pour alpha_1? 1
valeur initiale pour alpha_2? .0000001
pas pour alpha_2? .001
valeur finale pour alpha_2? .1
valeur initiale pour beta? 0
pas pour beta? .1
valeur finale pour beta? 1
```

```
point de depart? [.5 .0000001 .5]


ans =

   0.60000000000000   0.00100010000000   0.10000000000000
```

Ainsi nous obtenons une première approximation grossière de $\widehat{\theta}_n$. Utilisons une nouvelle fois la fonction *grille* en centrant les intervalles sur la valeur $(.6, .0010001, .1)$ et en reserrant les pas :

```
>> grille
quelle annee? 2003
simulations (1) ou observations reelles (2)? 2
valeur initiale pour alpha_1? .5
pas pour alpha_1? .01
valeur finale pour alpha_1? .7
valeur initiale pour alpha_2? .0000001
pas pour alpha_2? .0001
valeur finale pour alpha_2? .002
valeur initiale pour beta? 0
pas pour beta? .01
valeur finale pour beta? .2
point de depart? [.5 .0000001 0]

ans =

   0.70000000000000   0.00020010000000   0.03000000000000
```

Nous obtenons ainsi une valeur qui nous servira de point de départ pour l'utilisation de la fonction *lsqnonlin* :

```
lsqnonlin(@estim,[.7 .0002001 .03])

ans =

   0.70000000602863   0.00020259303691   0.02999997949082
```

Ainsi la valeur estimée des paramètres pour l'année 2003 est la suivante :

$$\widehat{\theta}_n = (.70000000602863, .00020259303691, .02999997949082)$$

Ce résultat a été obtenu en prenant comme surface maximale de bassin versant S_M=30 000 m^2 (surfmax=30000 dans le programme Matlab *haut.m*). La comparaison entre les hauteurs estimées avec ces paramètres et les hauteurs observées réelles nous a amené a recommencer ce travail d'estimation des paramètres, en prenant cette fois pour valeur S_M=50 000 m^2.

Nous avons alors obtenu

$$\widehat{\theta}_n = (.81000000250659, .00100042019713, .11999999529067)$$

Les graphiques suivants permettent de comparer les observations réelles et les hauteurs simulées avec les paramètres estimés précédemment, dans un premier temps en ayant pris $S_M = 30000$, puis avec $S_M = 50000$. La deuxième simulation correspondant plus à la réalité, nous garderons à l'avenir cette surface de bassin versant de 50 000 m^2 (NB : sauf mention contraire, tous les graphiques précédents étaient réalisés avec une surface S_m=30 000 m^2).

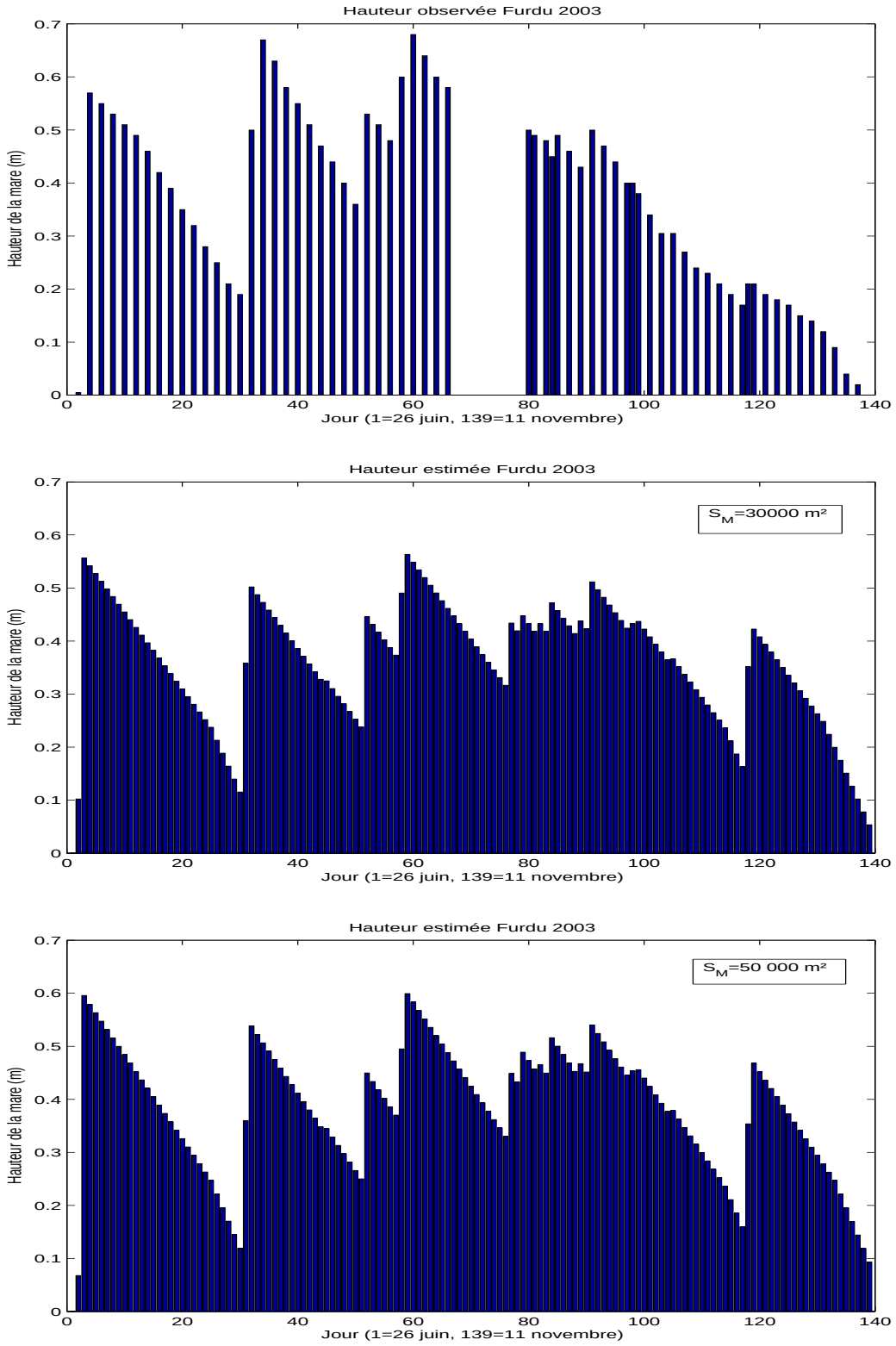


FIG. 6.4 – Comparaison entre les observations réelles et la simulation avec les paramètres estimés, pour une surface de bassin versant de 30 000 m^2 puis de 50 000 m^2

Chapitre 7

Programmes en Matlab

7.1 Programmation du modèle

```
function H=haut(x,a,imprim);
%x vecteur de parametres (longueur 3)
%simulation des hauteurs de l'annee a
%retourne un vecteur de longueur n
%et en option (si imprim=1) sort le graphique

%initialisation en fonction de l'annee
if a==2001
    n=103;
    h_init=.47;
    %pour la sortie graphique :
    s_0='1 juillet';
    S_1='11 octobre';
elseif a==2002
    n=117;
    h_init=.07;
    s_0='7 juillet';
    S_1='31 octobre';
elseif a==2003
    n=139;
    h_init=0;
    s_0='26 juin';
    s_1='11 novembre';
end

%debut du calcul
h_max=.7;
```

```
surf_fond=9;
surf_max=50000;

h_evap=.25;
E1=.02;
E2=.01;
a1=x(1);
a2=x(2);
B=x(3);


H=zeros(1,n);

for j=1:n

if j<=1
    H(j)=h_init;

else
    %calcul de h_bef
    if pluie(a,j)==0
        h_bef=H(j-1); %pas de pluie directe ni de ruissellement
    else
        %calcul de delta_ruiss

        V=pluie(a,j)*(1-infiltr(a,j,a1,a2,B))*(surf_max-surfce(H(j-1)));
             %volume d'eau qui ruisselle sur les bords

        if H(j-1)>0
            P=palier(H(j-1));
            v=surfce(H(j-1))*(P-H(j-1));

            if V<v
                delta_ruiss=V/surfce(H(j-1));
            else
                k=0;
                while v<=V
                    v=v+surfce(P+k*.05)*.05;
                    k=k+1;
                end
                k=k-1;
                v=v-surfce(P+k*.05)*.05;
                delta_ruiss=(V-v)/surfce(P+k*.05)+P+k*.05-H(j-1);
```

```
        end

    else
        %la hauteur est nulle, donc on travaille
        %avec la surface du fond
        v=surf_fond*.05;

        if V<v
            delta_ruiss=V/surf_fond;
        else
            k=1;
            while v<=V
                v=v+surfce(k*.05)*.05;
                k=k+1;
            end
            k=k-1;
            v=v-surfce(k*.05)*.05;
            delta_ruiss=(V-v)/surfce(k*.05)+k*.05;
        end
    end
    %fin de calcul de delta_ruiss

    h_bef=H(j-1)+pluie(a,j)+delta_ruiss;
end


%calcul de delta_evap
if h_bef>h_evap
    delta_evap=E2;
else
    delta_evap=E1;
end

%calcul de delta_infiltr
z=zeros(1,j-1);
ind = H==0;
ind2 = find(ind(:,1:1:j-1));
ind3 = H>0;
ind4 = find(ind3(:,1:1:j-1));
z(ind2)=pluie(a,ind2).*coeff(B,j,ind2);
z(ind4)=H(ind4).*coeff(B,j,ind4);
c=a1*a2/(a2+sum(z)); %calcul de alpha_tilde(j)
delta_infiltr=h_bef*c;
```

```
    %calcul de h_tilde(j)=h_aft
    h_aft=h_bef-delta_evap-delta_infiltr;

    %calcul de h(j)
    if h_aft>h_max
        H(j)=h_max;
    elseif h_aft<=0
        H(j)=0;
    else
        H(j)=h_aft;
    end

end
end

if imprim==1
    bar(H)
    title(sprintf('Hauteur simulee Furdu %d',a))
    xlabel(sprintf('Jour (1=%s, %d=%s)',s_0,n,s_1))
    ylabel('Hauteur de la mare (m)')
end
```

7.2 Fonctions appelées par le modèle

7.2.1 La fonction *pluie*

```
function p=pluie(a,j)

%pluie au jour j et annee a

if a==2001
    p=pluie2001(j);
elseif a==2002
    p=pluie2002(j);
elseif a==2003
    p=pluietriche2003(j);
end
```

où les fonctions *pluie200..* ont été programmées à partir des données, et retourne

la pluviométrie (en m) du jour j (voir **Données pluviométriques**). La fonction *pluietriche2003* simule artificiellement le premier pic de pluie, qui n'était pas reproduit dans les données initiales.

7.2.2 La fonction *surfce*

La fonction *surfce* fournit la surface de la mare (ici de Furdu) en fonction de la hauteur. Elle utilise les données de *surfpaliers* (reproduites dans le chapitre **Données - Surface de la mare**) qui est une fonction en escalier fournissant la surface par pas de hauteur de 5 cm. L'approximation des données manquantes se fait grâce à la relation de Thalès (cf **Calcul de** $\Delta^{ruiss}h_i$).

```
function s=surfce(h)
%surface en fonction de hauteur
%approximation avec Thales

H1=palier(h)-.05;
S1=surfpaliers(H1);
H2=palier(h);
S2=surfpaliers(H2);

s=((S2-S1)*(h-H1)/.5)+S1;



function H=palier(h)
%retourne la hauteur immediatement superieure a  h
%et multiple de 0.05
%(donc correspondant a la division par pas de 5 cm)

H=0;
while h>H
    H=H+.05;
end
```

7.2.3 La fonction *coeff*

Elle calcule le coefficient $a_{i,j}$ traduisant l'influence du jour j sur le jour i, en fonction du paramètre β.

```
function a=coeff(B,i,j)
%coeff en fontion du parametre B

a=B.^(i-j);
```

7.2.4 La fonction *infiltr*

Elle calcule le coefficient de saturation du sol en eau $\widetilde{\alpha}(i)$.

```
function b=infiltr(a,j,a1,a2,B)
%infiltr en fonction des parametres (a1,a2,B)
%pour l'annee a et le jour j

x=[];

for k=1:j-1
    x= [x pluie(a,k).*coeff(B,j,k)];
end

b=a1*a2/(a2+sum(x));
```

7.3 Estimation des paramètres

7.3.1 La fonction *moindrcarr*

Elle calcule

$$S_n(\theta) = \sum_{i=1}^{n} (h^{obs}(i) - h_\theta(i))^2$$

en prenant en entrée, sous forme d'un *vecteur* de longueur n, les observations $h^{obs}(i)$. Celles-ci peuvent être fournies par une simulation (on rentre alors *choix*=1) ou bien par des observations réelles (*choix*=2). Cette distinction permet, en cas d'observations réelles, de demander au programme de ne pas tenir compte des journées non observées.

Elle prend également en entrée l'*année* et le vecteur de paramètres de longueur 3 *param*, ce qui permettra de calculer les $h_\theta(i)$ pour l'année demandée et pour $\theta = param$.

```
function m=moindrcarr(choix,annee,vecteur,param)

%en entree:

%  - choix (1 si simulations ou 2 si observations reelles)
%    si choix=2 supprimer les journees non observees

%  - annee
%    pour la simulation

%  - vecteur (de longueur n)
```

```
%    soit celui des observations reelles,
%    soit obtenu par une simulation

%  - parametre
%    pour la simulation

if annee==2001
    n=103;
elseif annee==2002
    n=117;
elseif annee==2003
    n=139;
end

z=zeros(1,n);
H2=haut(param,annee,0);

if choix==2
    ind=find(vecteur>0); %enleve journees non observees
    z(ind)=(vecteur(ind) - H2(ind)).^2;

else
    for k=1:n
        z(k)=(vecteur(k) - H2(k)).^2;
    end
end

m=sum(z);
```

7.3.2 La fonction *grille*

Etape préliminaire à l'utilisation de la fonction Matlab *lsqnonlin* (cf **Estimation des paramètres : deux étapes**) : permet d'obtenir une valeur assez proche de

$$\widehat{\theta}_n = \text{argmin}_\theta \, S_n(\theta)$$

```
function theta=grille()

%retourne le argmin des valeurs prises par
%'somme des carres' (moindrcarr.m) sur une grille

a=input('quelle annee?');

choix=input('simulations (1) ou observations reelles (2)?');
```

```
if choix==1
    par_sim=input('parametre pour la simulation?');
    H1=haut(par_sim,a,0);
else
    H1=hautobs(a);
    ind=find(H1>0);
    %pour ne pas tenir compte des jours non observes
end

%construction de la grille
a1_init=input('valeur initiale pour alpha_1?');
a1_pas=input('pas pour alpha_1?');
a1_fin=input('valeur finale pour alpha_1?');
a2_init=input('valeur initiale pour alpha_2?');
a2_pas=input('pas pour alpha_2?');
a2_fin=input('valeur finale pour alpha_2?');
b_init=input('valeur initiale pour beta?');
b_pas=input('pas pour beta?');
b_fin=input('valeur finale pour beta?');

%point de depart pour calculer la somme en ce point
%et comparer aux autres valeurs
%pas besoin d'etre sur la grille
theta=input('point de depart?');

%calcul de la somme en ce point
S=moindrcarr(choix,a,H1,theta);

%calcul en tous points de la grille et comparaison
for j=a1_init:a1_pas:a1_fin
    for k=a2_init:a2_pas:a2_fin
        for l=b_init:b_pas:b_fin

            if moindrcarr(choix,a,H1,[j k l])<S
                S=moindrcarr(choix,a,H1,[j k l]);
                theta=[j k l];
            end

        end
    end
end
```

7.3.3 Utilisation de *lsqnonlin*

Pour utiliser *lsqnonlin* sur

$$S_n(\theta) = \sum_{i=1}^{n} (h^{obs}(i) - h_\theta(i))^2$$

il faut traduire cette fonction $S_n(\theta)$ comme une fonction à valeur vectorielle, sans tenir compte de la somme ni mettre les éléments au carré :

$$estim(\theta) = \begin{bmatrix} h^{obs}(1) - h_\theta(1) \\ \dots \\ h^{obs}(n) - h_\theta(n) \end{bmatrix}$$

La fonction *lsqnonlin* calculera ensuite d'elle-même

$$\widehat{\theta}_n = \text{argmin}_\theta \sum_{i=1}^{n} (h^{obs}(i) - h_\theta(i))^2$$

en prenant en entrée un paramètre θ_1.

```
function E=estim(x)
% x vecteur de longueur 3
% retourne vecteur de longueur n

a=2003;

if a==2001
    n=103;
elseif a==2002
    n=117;
elseif a==2003
    n=139;
end

E=zeros(1,n);
H1=hautobs(a); %ou eventuellement des hauteurs simulees
               %par exemple H1=haut([.9 .001 .7],a,0)
H2=haut(x,a,0);

ind=find(H1>0); %pour ne pas considerer les jours non observes
                 %(pas besoin si simulation)
E(ind)=H1(ind)-H2(ind);
```

On appelle ensuite ainsi la fonction :

$$\textit{lsqnonlin}\ (@\text{estim}\ ,\ \theta_1)$$

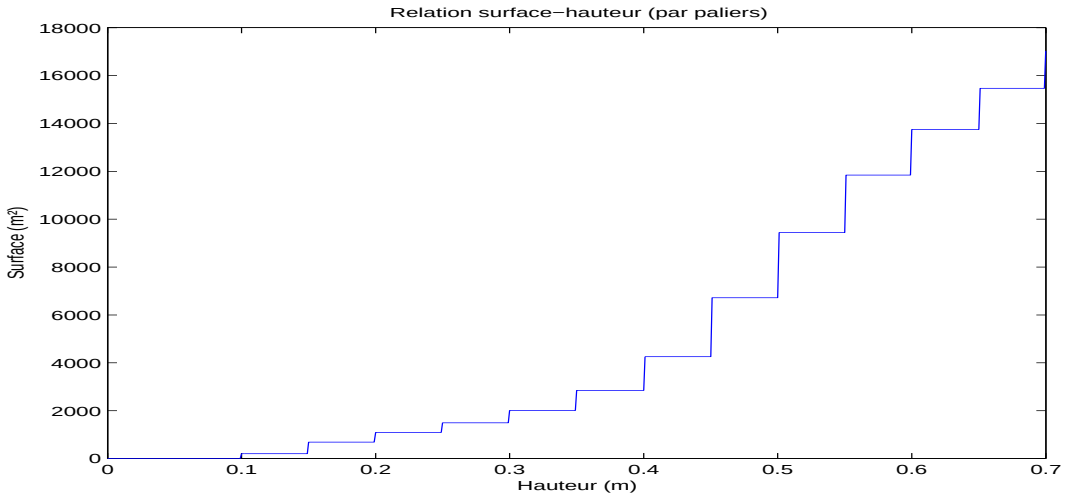
et on obtient en retour une valeur de $\widehat{\theta}_n$.

Chapitre 8

Données

8.1 Surface de la mare

```
function s=surfpaliers(x)

%retourne vecteur de meme longueur que x
%donne surface en fonction de la hauteur
% par pas de 5 cm

s=zeros(1,length(x));

for j=1:length(x)

if x(j)>=0 & x(j)<.05
   s(j)=0;
elseif x(j)>=.05 & x(j)<.1
   s(j)=9;
elseif x(j)>=.1 & x(j)<.15
   s(j)=204;
elseif x(j)>=.15 & x(j)<.2
   s(j)=688;
elseif x(j)>=.2 & x(j)<.25
   s(j)=1091;
elseif x(j)>=.25 & x(j)<.3
   s(j)=1494;
elseif x(j)>=.3 & x(j)<.35
   s(j)=2008;
elseif x(j)>=.35 & x(j)<.4
   s(j)=2849;
elseif x(j)>=.4 & x(j)<.45
   s(j)=4254;
elseif x(j)>=.45 & x(j)<.5
   s(j)=6718;
elseif x(j)>=.5 & x(j)<.55
   s(j)=9438;
elseif x(j)>=.55 & x(j)<.6
   s(j)=11843;
elseif x(j)>=.6 & x(j)<.65
   s(j)=13748;
elseif x(j)>=.65 & x(j)<.7
   s(j)=15468;
else
   s(j)=17030;
end
end
```



8.2 Données pluviométriques

8.2.1 Année 2001

```
function p=pluie2001(i,imprim)

%donnees de T_pluvio.txt de 1juill a  11oct
%retourne la pluviometrie du jour i
%(si i est un vecteur, retourne vecteur p de meme taille
%commodite pour utilisation dans programme Matlab)

%faire pluie2001(103,1) pour sortie graphique de l'annee

p=zeros(size(i));

for j=1:length(i)

if i(j)==4
   p(j)=.0416;
elseif i(j)==6
   p(j)=.0063;
elseif i(j)==13
   p(j)=.0017;
elseif i(j)==18
   p(j)=.005;
elseif i(j)==19
   p(j)=.0094;
elseif i(j)==26
   p(j)=.01610;
elseif i(j)==27
   p(j)=.0092;
elseif i(j)==28
   p(j)=.007;
elseif i(j)==31
   p(j)=.0025;
elseif i(j)==32
   p(j)=.0025;
elseif i(j)==38
   p(j)=.0449;
elseif i(j)==41
   p(j)=.0305;
elseif i(j)==44
```

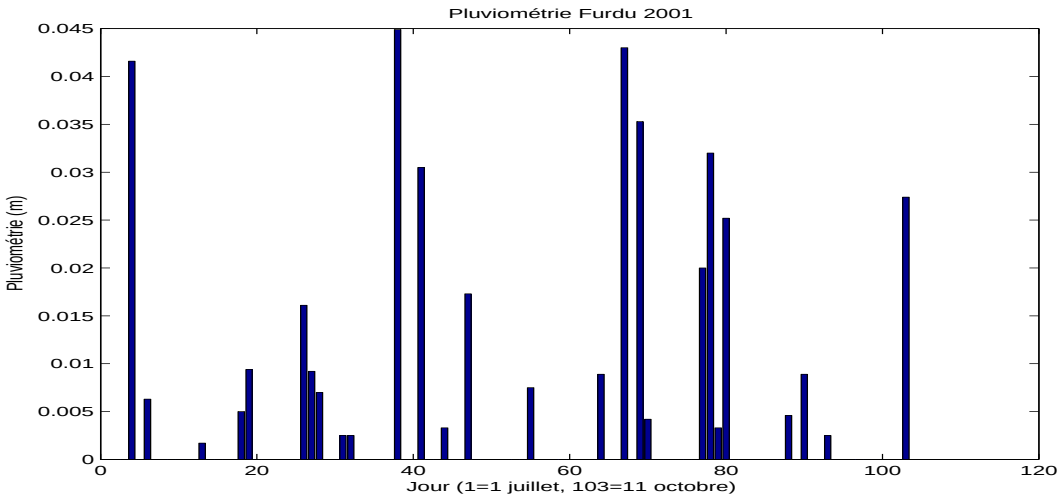
```
   p(j)=.0033;
elseif i(j)==47
   p(j)=.0173;
elseif i(j)==55
   p(j)=.0075;
elseif i(j)==64
   p(j)=.0089;
elseif i(j)==67
   p(j)=.043;
elseif i(j)==69
   p(j)=.0353;
elseif i(j)==70
   p(j)=.0042;
elseif i(j)==77
   p(j)=.02;
elseif i(j)==78
   p(j)=.032;
elseif i(j)==79
   p(j)=.0033;
elseif i(j)==80
   p(j)=.0252;
elseif i(j)==88
   p(j)=.0046;
elseif i(j)==90
   p(j)=.0089;
elseif i(j)==93
   p(j)=.0025;
elseif i(j)==103
   p(j)=.0274;
else
   p(j)=0;
end

end

if imprim==1
    P=zeros(1,103);
    for k=1:103
        P(k)=pluie2001(k,0);
    end
    bar(P)
    title('Pluviometrie Furdu 2001')
    xlabel('Jour (1=1 juillet, 103=11 octobre)')
    ylabel('Pluviometrie (m)')
end

\subsection{Année 2002}
\scriptsize
\begin{verbatim}
function p=pluie2002(i,imprim)

%donnees de T_pluvio.txt de 7juill a  31oct
%retourne la pluviometrie du jour i
%(si i est un vecteur, retourne vecteur p de meme taille)

%faire pluie2002(117,1) pour sortie graphique de l'annee

p=zeros(size(i));

for j=1:length(i)

if i(j)==1
   p(j)=.0208;
elseif i(j)==18
   p(j)=.0176;
```

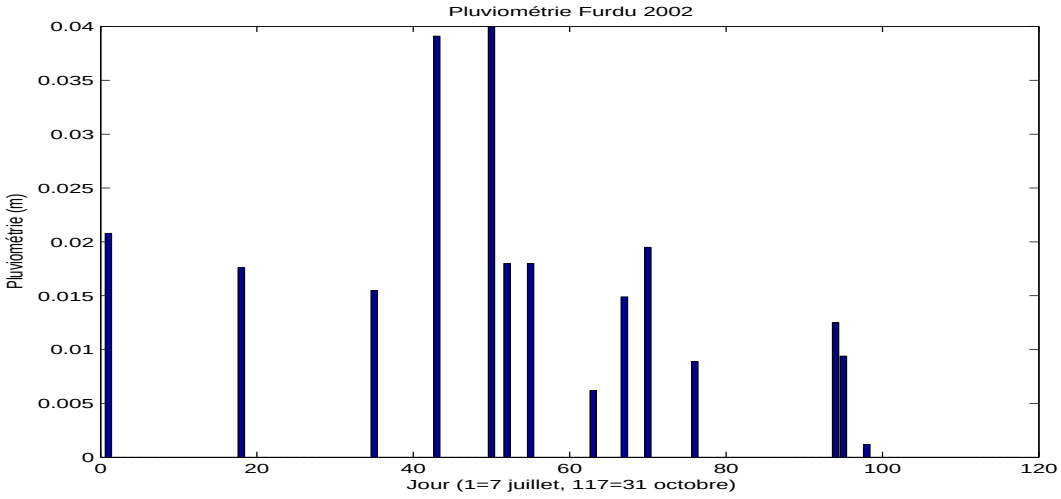
```
elseif i(j)==35
   p(j)=.0155;
elseif i(j)==43
   p(j)=.0391;
elseif i(j)==50
   p(j)=.04;
elseif i(j)==52
   p(j)=.018;
elseif i(j)==55
   p(j)=.018;
elseif i(j)==63
   p(j)=.0062;
elseif i(j)==67
   p(j)=.0149;
elseif i(j)==70
   p(j)=.0195;
elseif i(j)==76
   p(j)=.0089;
elseif i(j)==94
   p(j)=.0125;
elseif i(j)==95
   p(j)=.0094;
elseif i(j)==98
   p(j)=.0012;
else
   p(j)=0;
end

end

if imprim==1
    P=zeros(1,117);
    for k=1:117
        P(k)=pluie2002(k,0);
    end
    bar(P)
    title('Pluviometrie Furdu 2002')
    xlabel('Jour (1=7 juillet, 117=31 octobre)')
    ylabel('Pluviometrie (m)')
end
```

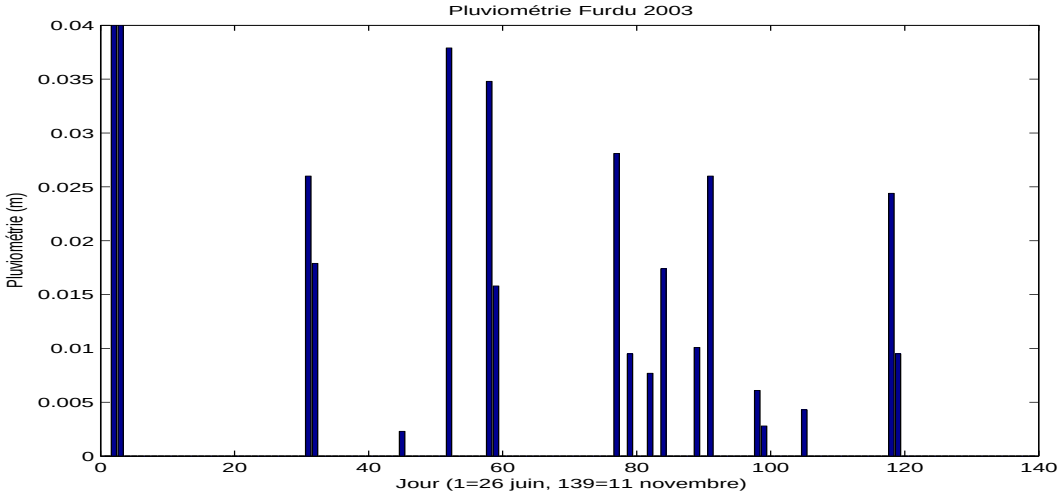
8.2.2 Année 2003

```
function p=pluietriche2003(i,imprim)

%donnees (+ triche) de pluvio2003.txt
%de 26juin a  11nov (triche de 26 juin Ã  28 juin)
%retourne la pluviometrie du jour i
%(si i est un vecteur, retourne vecteur p de meme taille)

%faire pluie2003(139,1) pour sortie graphique de l'annee

p=zeros(size(i));

for j=1:length(i)

if i(j)==2
    p(j)=.04;
elseif i(j)==3
    p(j)=.04;
elseif i(j)==31
   p(j)=.026;
elseif i(j)==32
```

```
   p(j)=.0179;
elseif i(j)==45
   p(j)=0.0023;
elseif i(j)==52
   p(j)=0.0379;
elseif i(j)==58
   p(j)=0.0348;
elseif i(j)==59
   p(j)=0.0158;
elseif i(j)==77
   p(j)=0.0281;
elseif i(j)==79
   p(j)=0.0095;
elseif i(j)==82
   p(j)=0.0077;
elseif i(j)==84
   p(j)=0.0174;
elseif i(j)==89
   p(j)=0.0101;
elseif i(j)==91
   p=0.026;
elseif i(j)==98
   p(j)=0.0061;
elseif i(j)==99
   p(j)=0.0028;
elseif i(j)==105
   p(j)=0.0043;
elseif i(j)==118
   p(j)=0.0244;
elseif i(j)==119
   p(j)=0.0095;
else
   p(j)=0;
end

end

if imprim==1
    P=zeros(1,139);
    for k=1:139
        P(k)=pluie2003(k,0);
    end
    bar(P)
    title('Pluviometrie Furdu 2003')
    xlabel('Jour (1=26 juin, 139=11 novembre)')
    ylabel('Pluviometrie (m)')
end
```


Pluviométrie Furdu 2001
Pluviométrie (m)
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
20
40
60
80
100
120
Jour (1=1 juillet, 103=11 octobre)


Pluviométrie Furdu 2002
Pluviométrie (m)
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
20
40
60
80
100
120
Jour (1=7 juillet, 117=31 octobre)


Pluviométrie Furdu 2003
Pluviométrie (m)
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
20
40
60
80
100
120
140
Jour (1=26 juin, 139=11 novembre)

8.3 Données limnimétriques

8.3.1 La fonction *hautobs*

Elle retourne les hauteurs observées pour l'année donnée en entrée, en utilisant les données répertoriées dans les fonctions *hautobs200...*

```
function H=hautobs(a)

%retourne vecteur de longueur n des hauteurs
%observees (a Furdu) pour l'annee a

if a==2001
    H=hautobs2001(103,0);

elseif a==2002
    H=hautobs2002(117,0);

elseif a==2003
    H=hautobstriche2003(139,0);

end
```

8.3.2 Année 2001

```
function h=hautobs2001(i,imprim)

%donnees T_limni.txt
%donne hauteur observee (m) jusqu'au jour i
%retourne vecteur ligne de longueur i
%si imprim=1, retourne graphique

h=zeros(1,i);

for j=1:i
    if j==4
        h(j)=.47;
    elseif j==6
        h(j)=.36;
    elseif j==11
        h(j)=.28;
    elseif j==12
        h(j)=.26;
    elseif j==13
        h(j)=.25;
    elseif j==14
        h(j)=.23;
    elseif j==15
        h(j)=.22;
    elseif j==16
        h(j)=.22;
    elseif j==17
        h(j)=.21;
    elseif j==18
```

```
    h(j)=.2;
elseif j==19
    h(j)=.2;
elseif j==20
    h(j)=.19;
elseif j==21
    h(j)=.18;
elseif j==22
    h(j)=.17;
elseif j==23
    h(j)=.16;
elseif j==24
    h(j)=.15;
elseif j==25
    h(j)=.14;
elseif j==26
    h(j)=.11;
elseif j==27
    h(j)=.12;
elseif j==28
    h(j)=.12;
elseif j==29
    h(j)=.13;
elseif j==30
    h(j)=.12;
elseif j==31
    h(j)=.11;
elseif j==32
    h(j)=.11;
elseif j==33
    h(j)=.1;
elseif j==34
    h(j)=.09;
elseif j==35
    h(j)=.09;
elseif j==36
    h(j)=.08;
elseif j==37
    h(j)=.08;
elseif j==38
    h(j)=.07;
elseif j==39
    h(j)=.09;
elseif j==40
    h(j)=.08;
elseif j==41
    h(j)=.07;
elseif j==42
    h(j)=.47;
elseif j==43
    h(j)=.45;
elseif j==44
    h(j)=.44;
elseif j==45
    h(j)=.43;
elseif j==46
    h(j)=.39;
elseif j==47
    h(j)=.4;
elseif j==48
    h(j)=.4;
elseif j==49
    h(j)=.38;
elseif j==50
    h(j)=.36;
```

```
elseif j==51
    h(j)=.34;
elseif j==52
    h(j)=.32;
elseif j==53
    h(j)=.3;
elseif j==54
    h(j)=.29;
elseif j==55
    h(j)=.28;
elseif j==56
    h(j)=.27;
elseif j==57
    h(j)=.26;
elseif j==58
    h(j)=.25;
elseif j==59
    h(j)=.24;
elseif j==60
    h(j)=.23;
elseif j==61
    h(j)=.22;
elseif j==62
    h(j)=.2;
elseif j==63
    h(j)=.18;
elseif j==64
    h(j)=.22;
elseif j==65
    h(j)=.2;
elseif j==66
    h(j)=.19;
elseif j==67
    h(j)=.31;
elseif j==68
    h(j)=.29;
elseif j==69
    h(j)=.40;
elseif j==70
    h(j)=.38;
elseif j==71
    h(j)=.47;
elseif j==72
    h(j)=.45;
elseif j==73
    h(j)=.43;
elseif j==74
    h(j)=.41;
elseif j==75
    h(j)=.39;
elseif j==76
    h(j)=.37;
elseif j==77
    h(j)=.34;
elseif j==78
    h(j)=.36;
elseif j==79
    h(j)=.39;
elseif j==80
    h(j)=.38;
elseif j==81
    h(j)=.59;
elseif j==82
    h(j)=.58;
elseif j==83
```

```
        h(j)=.56;
    elseif j==84
        h(j)=.54;
    elseif j==85
        h(j)=.52;
    elseif j==86
        h(j)=.50;
    elseif j==87
        h(j)=.49;
    elseif j==88
        h(j)=.48;
    elseif j==89
        h(j)=.46;
    elseif j==90
        h(j)=.44;
    elseif j==91
        h(j)=.43;
    elseif j==92
        h(j)=.41;
    elseif j==94
        h(j)=.37;
    elseif j==101
        h(j)=.28;
    elseif j==102
        h(j)=.27;
    elseif j==103
        h(j)=.31;
    else
        h(j)=0;
    end
end

if imprim==1
    bar(h)
    title('Hauteur observee Furdu 2001')
    xlabel('Jour (1=1 juillet, 103=11 octobre)')
    ylabel('Hauteur de la mare (m)')
end
```

8.3.3 Année 2002

```
function h=hautobs2001(i,imprim)

%donnees T_limni.txt
%donne hauteur observÃ©e (m) jusqu'au jour i
%retourne vecteur ligne de longueur i
%si imprim=1, retourne graphique

h=zeros(1,i);

for j=1:i
    if j==1
        h(j)=.07;
    elseif j==2
        h(j)=.05;
    elseif j==3
        h(j)=.04;
    elseif j==4
        h(j)=.02;
    elseif j==20
        h(j)=.02;
    elseif j==38
        h(j)=.4;
```

```
elseif j==39
    h(j)=.35;
elseif j==40
    h(j)=.33;
elseif j==41
    h(j)=.31;
elseif j==42
    h(j)=.29;
elseif j==43
    h(j)=.27;
elseif j==44
    h(j)=.45;
elseif j==45
    h(j)=.4;
elseif j==46
    h(j)=.38;
elseif j==47
    h(j)=.34;
elseif j==48
    h(j)=.32;
elseif j==49
    h(j)=.31;
elseif j==50
    h(j)=.3;
elseif j==51
    h(j)=.5;
elseif j==52
    h(j)=.45;
elseif j==53
    h(j)=.5;
elseif j==54
    h(j)=.48;
elseif j==55
    h(j)=.46;
elseif j==56
    h(j)=.55;
elseif j==57
    h(j)=.55;
elseif j==58
    h(j)=.54;
elseif j==59
    h(j)=.53;
elseif j==60
    h(j)=.51;
elseif j==61
    h(j)=.49;
elseif j==62
    h(j)=.47;
elseif j==63
    h(j)=.45;
elseif j==64
    h(j)=.43;
elseif j==65
    h(j)=.41;
elseif j==66
    h(j)=.39;
elseif j==67
    h(j)=.42;
elseif j==68
    h(j)=.4;
elseif j==69
    h(j)=.39;
elseif j==70
    h(j)=.42;
elseif j==71
```

```
    h(j)=.38;
elseif j==72
    h(j)=.37;
elseif j==73
    h(j)=.34;
elseif j==74
    h(j)=.32;
elseif j==75
    h(j)=.33;
elseif j==76
    h(j)=.32;
elseif j==77
    h(j)=.31;
elseif j==78
    h(j)=.29;
elseif j==79
    h(j)=.28;
elseif j==80
    h(j)=.27;
elseif j==81
    h(j)=.26;
elseif j==82
    h(j)=.25;
elseif j==83
    h(j)=.24;
elseif j==84
    h(j)=.23;
elseif j==85
    h(j)=.21;
elseif j==86
    h(j)=.19;
elseif j==87
    h(j)=.17;
elseif j==88
    h(j)=.15;
elseif j==89
    h(j)=.13;
elseif j==90
    h(j)=.12;
elseif j==91
    h(j)=.11;
elseif j==92
    h(j)=.1;
elseif j==93
    h(j)=.11;
elseif j==94
    h(j)=.35;
elseif j==95
    h(j)=.34;
elseif j==96
    h(j)=.33;
elseif j==97
    h(j)=.32;
elseif j==98
    h(j)=.3;
elseif j==99
    h(j)=.29;
elseif j==100
    h(j)=.28;
elseif j==101
    h(j)=.27;
elseif j==102
    h(j)=.26;
elseif j==103
    h(j)=.25;
```

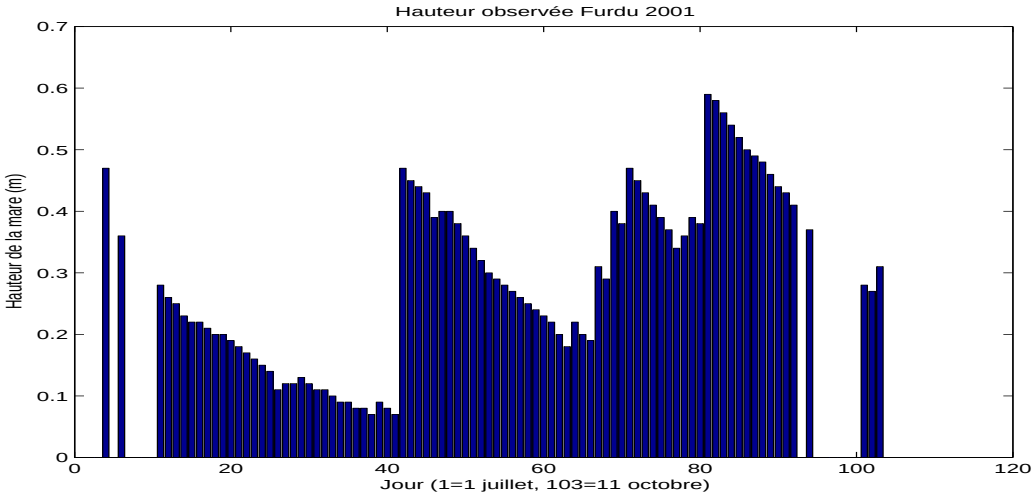
```
    elseif j==104
        h(j)=.23;
    elseif j==105
        h(j)=.21;
    elseif j==106
        h(j)=.19;
    elseif j==107
        h(j)=.17;
    elseif j==108
        h(j)=.14;
    elseif j==109
        h(j)=.11;
    elseif j==110
        h(j)=.09;
    elseif j==111
        h(j)=.06;
    elseif j==112
        h(j)=.04;
    elseif j==113
        h(j)=.02;
    else
        h(j)=0;
    end
end

if imprim==1
    bar(h)
    title('Hauteur observee Furdu 2002')
    xlabel('Jour (1=7 juillet, 117=31 octobre)')
    ylabel('Hauteur de la mare (m)')
end
```

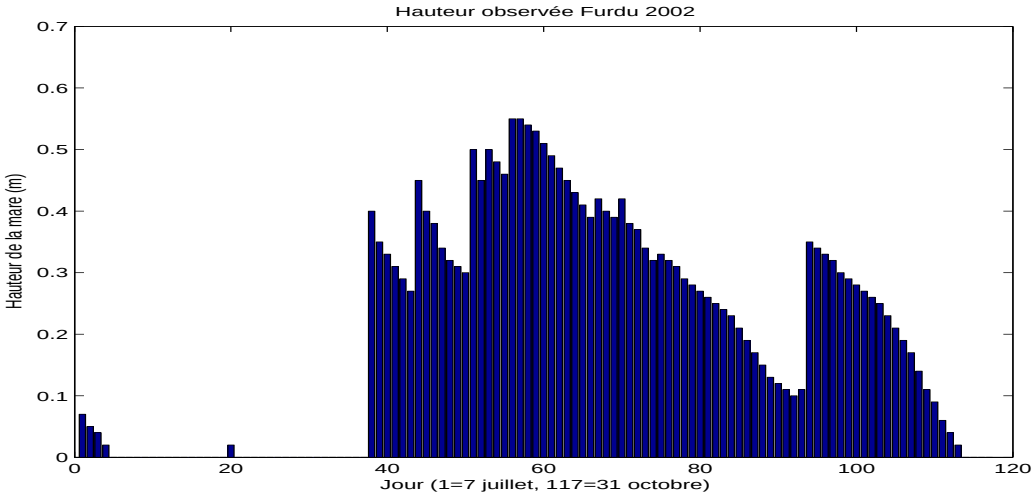
8.4 Année 2003

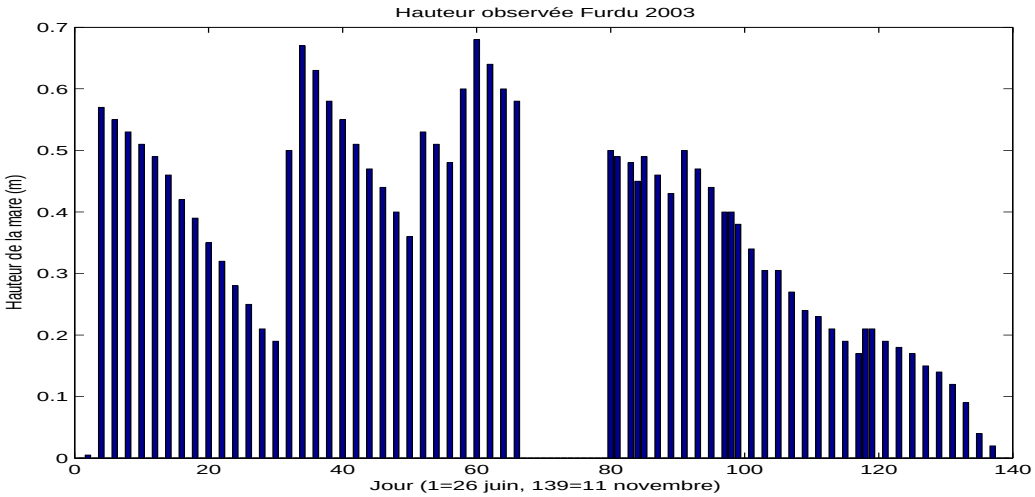
```
function h=hautobstriche2003(i,imprim)

%donnees limni2003.xls
%donne hauteur observee (m) jusqu'au jour i
%retourne vecteur ligne de longueur i
%du 26 juin au 11 novembre
%triche du 26 juin au 7 juillet (simulation de la premiere pluie)
%si imprim=1, retourne graphique

h=zeros(1,i);

for j=1:i
    if j==2
        h(j)=.005;
    elseif j==4
        h(j)=.57;
    elseif j==6
        h(j)=.55;
    elseif j==8
        h(j)=.53;
    elseif j==10
        h(j)=.51;
    elseif j==12
        h(j)=.49;
    elseif j==14
        h(j)=.46;
    elseif j==16
        h(j)=.42;
    elseif j==17
```

```
    h(j)=.39;
elseif j==20
    h(j)=.35;
elseif j==22
    h(j)=.32;
elseif j==24
    h(j)=.28;
elseif j==26
    h(j)=.25;
elseif j==28
    h(j)=.21;
elseif j==30
    h(j)=.19;
elseif j==32
    h(j)=.5;
elseif j==34
    h(j)=.67;
elseif j==36
    h(j)=.63;
elseif j==38
    h(j)=.58;
elseif j==40
    h(j)=.55;
elseif j==42
    h(j)=.51;
elseif j==44
    h(j)=.47;
elseif j==46
    h(j)=.44;
elseif j==48
    h(j)=.4;
elseif j==50
    h(j)=.36;
elseif j==52
    h(j)=.53;
elseif j==54
    h(j)=.51;
elseif j==56
    h(j)=.48;
elseif j==58
    h(j)=.6;
elseif j==60
    h(j)=.68;
elseif j==62
    h(j)=.64;
elseif j==64
    h(j)=.6;
elseif j==66
    h(j)=.58;
elseif j==80
    h(j)=.5;
elseif j==81
    h(j)=.49;
elseif j==83
    h(j)=.48;
elseif j==84
    h(j)=.45;
elseif j==85
    h(j)=.49;
elseif j==87
    h(j)=.46;
elseif j==89
    h(j)=.43;
elseif j==91
    h(j)=.5;
```

```
    elseif j==93
        h(j)=.47;
    elseif j==95
        h(j)=.44;
    elseif j==97
        h(j)=.4;
    elseif j==98
        h(j)=.4;
    elseif j==99
        h(j)=.38;
    elseif j==101
        h(j)=.34;
    elseif j==103
        h(j)=.305;
    elseif j==105
        h(j)=.305;
    elseif j==107
        h(j)=.27;
    elseif j==109
        h(j)=.24;
    elseif j==111
        h(j)=.23;
    elseif j==113
        h(j)=.21;
    elseif j==115
        h(j)=.19;
    elseif j==117
        h(j)=.17;
    elseif j==118
        h(j)=.21;
    elseif j==119
        h(j)=.21;
    elseif j==121
        h(j)=.19;
    elseif j==123
        h(j)=.18;
    elseif j==125
        h(j)=.17;
    elseif j==127
        h(j)=.15;
    elseif j==129
        h(j)=.14;
    elseif j==131
        h(j)=.12;
    elseif j==133
        h(j)=.09;
    elseif j==135
        h(j)=.04;
    elseif j==137
        h(j)=.02;
    else
        h(j)=0;
    end
end
if imprim==1
    bar(h)
    title('Hauteur observee Furdu 2003')
    xlabel('Jour (1=26 juin, 139=11 novembre)')
    ylabel('Hauteur de la mare (m)')
end
```


Hauteur observée Furdu 2001
Hauteur de la mare (m)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
20
40
60
80
100
120
Jour (1=1 juillet, 103=11 octobre)


Hauteur observée Furdu 2002
Hauteur de la mare (m)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
20
40
60
80
100
120
Jour (1=7 juillet, 117=31 octobre)


Hauteur observée Furdu 2003
Hauteur de la mare (m)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
20
40
60
80
100
120
140
Jour (1=26 juin, 139=11 novembre)