BERTHE Hélène
WATREMEZ Alexis
HMIMINA Gabriel

Étude de l’absence d’assainissement sur les concentrations en ions azotés au sein de la nappe perchée de Coucy-les-Eppes.

2003-2004

Introduction

Coucy-les-Eppes est une commune rurale de six cent trente habitants située à une quinzaine de kilomètres à l'est de Laon, dans l'Aisne. Elle comporte une nappe phréatique affleurant en certains points, et la faible profondeur de cette nappe associée à la nature sableuse du sol sur lequel le village est implanté rend l'implantation de fosses septiques inappropriée.
Aussi le village, majoritairement constitué d'habitations anciennes, qui ne possède pas de système d'assainissement est riche en puits de perte ou puits perdus ; de simples puits inutilisés dans lesquels les déchets sont déversés.
Qui plus est, un ancien lavoir alimenté par une source dite de la Croix-de-Bois présente une prolifération de lentilles d'eau, et le ruissellement de l'eau de la nappe donne lieu rapidement à l'apparition d'algues filamenteuses vertes, ce qui laisse présager une pollution au nitrate pouvant être liée à un apport d'urée.
S'ensuit la problématique suivante : quel est le devenir, et quelles sont les conséquences de cet apport anthropique ? Y a t'il une pollution de la nappe, et si oui, comment pourrait-elle évoluer en fonction des aménagements possibles ?
Une réponse à cette question sera apportée dans un premier temps par la localisation et la caractérisation du terrain étudié qui sera informatiquement modélisé, puis par la caractérisation des apports anthropiques étudiés et la quantification de l'éventuelle pollution, et enfin par la modélisation du fonctionnement de la nappe phréatique.
Celle-ci permettra la localisation de zones d'accumulation des déchets azotés et la quantification du temps nécessaire au lessivage des nitrates et nitrites en l'absence d'apport.

Sommaire

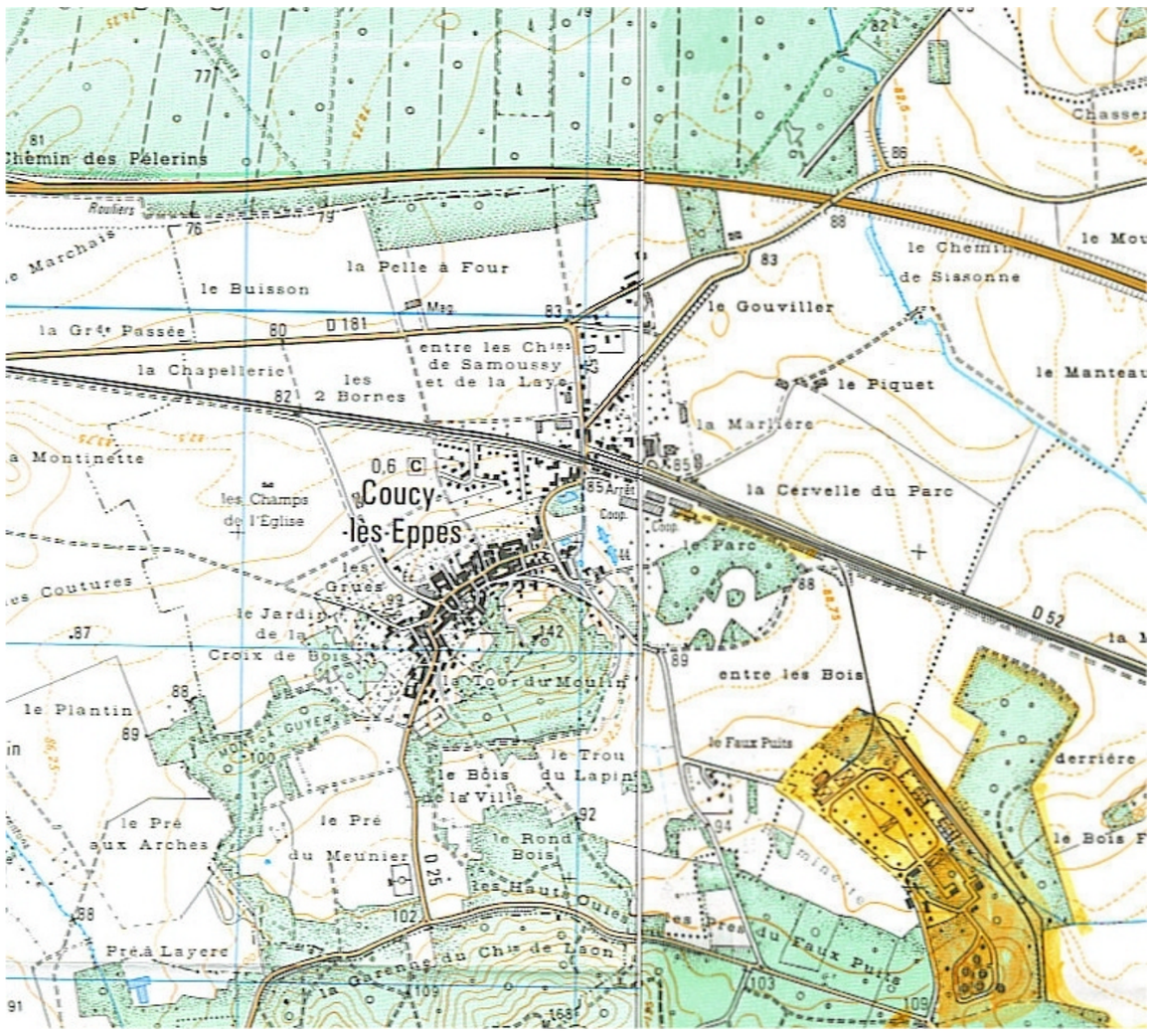
Synthèse de l'étude

I) Localisation et caractérisation du terrain étudié

A) Étude et modélisation de la topographie

1) Étude de la topographie et des sols à l'affleurement

Le village de Coucy est installé sur une butte de 142 mètres de haut essentiellement sableuse. On trouve à l'affleurement ponctuellement du grès, et une large bande argileuse.
L'orientation du réseau de routes du village favorise un ruissellement, principalement du sud vers le nord.
L'extrait de carte topographique suivant représente la zone étudiée :

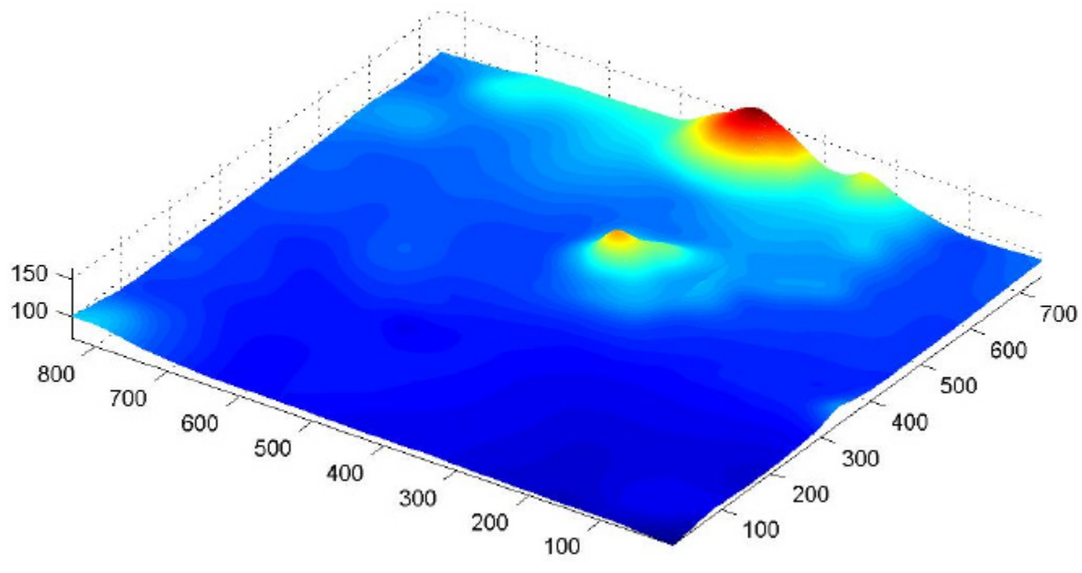


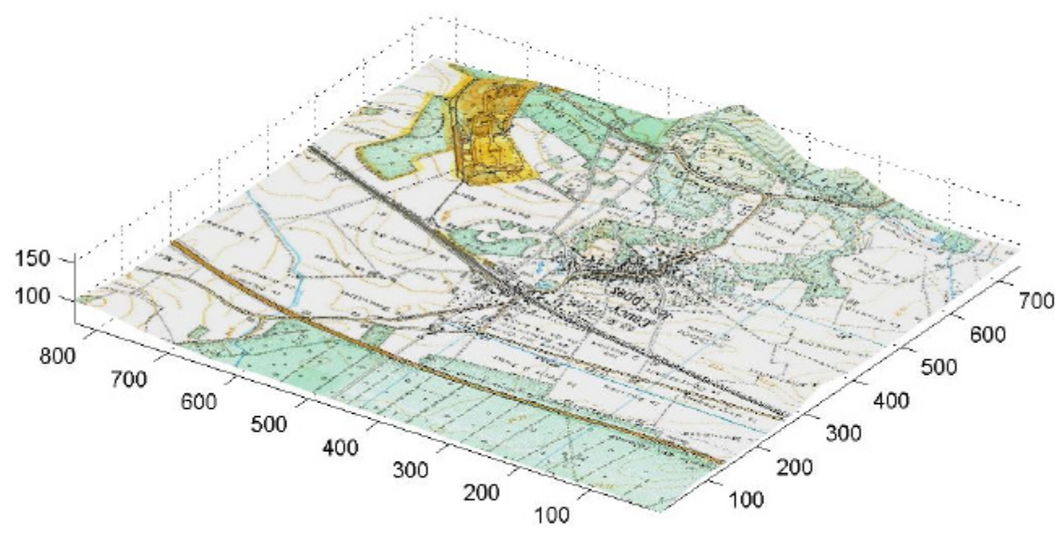
extrait de carte topographique, cf. référence (1)

Le relief au sud atteint dans la zone étudiée une altitude maximum de 165 mètres et présente une alternance d'affleurements sableux et argileux recouverts par des peuplements forestiers relativement denses.

2) Modélisation numérique de la topographie.

L'application de filtres graphiques permet de mettre en évidence les lignes de niveau que le logiciel Matlab acquiert ensuite sous la forme de coordonnées des pixels les constituant.
Une fonction permet ensuite d'extraire de la matrice des coordonnées les ensembles géographiques représentant les lignes de niveaux. La saisie des altitudes permet l'obtention d'un set de points de la carte d'altitudes connues, c'est à dire un set de « nœuds ».
La fonction *tpaps* de Matlab R13 renvoie la structure d'une surface passant au plus près des points connus grâce à une approximation polynomiale de degré 3.
Cette structure est ensuite convertie en matrice altitude décrivant le terrain grâce à une fonction *fnplt* de Matlab R13 modifiée à cette fin. Le résultat de ces opérations est le suivant :





L'erreur commise lors de l'approximation de la surface par des B-splines a pu être évaluée par la comparaison des altitudes lues et celles calculées au niveau de 16500 points. La valeur moyenne est de 0.37 m , avec une variance de 0.38 m².
L'obtention de cette matrice permet la modélisation de la topographie ainsi que la description du ruissellement en fonction de la nature des terrains et de la pluviométrie.

B) Mise en évidence de la présence d'une nappe et caractérisation de l'aquifère

Une simple observation sur le terrain permet la mise en évidence de nombreuses sources alimentées par une nappe phréatique à faible profondeur. Celle-ci est matérialisée par une ligne de sources coïncidant avec la limite entre l'affleurement de la couche sableuse sur laquelle Coucy-les-Eppes repose principalement et celui d'une couche argileuse.
De plus, en période pluvieuse, certaines caves sont inondées et le niveau de certains puits peu profonds augmente.

La nappe étudiée est donc localisée au sein des sables et grès de Bracheux, libre dans la zone de Coucy-les-Eppes et a pour mur les argiles de Vaux-sous-Laon.

C) Étude et modélisation de la structure géologique du terrain

1) Étude de la structure géologique du terrain.

Une étude d'un extrait de la carte géologique *(référence* **(2)***)* permet de mettre en évidence quelques caractéristiques :

- Les courbes de niveau suivent généralement les limites d'affleurement. La structure est donc approximativement *tabulaire.*
- Les terrains à l'affleurement sont essentiellement tertiaires.

Terrain	**Age**	**Nature**
- e4b	Yprésien supérieur (III)	*Argile* de Laon (7m au maximum)
- e4a	Yprésien supérieur (III)	Sables de Cuise (50 à 60 m)
- e3	Yprésien inférieur (III)	*Argiles* vertes ou grises
- e2c	Thanétien supérieur (III)	Sables et grés de Bracheux (25 à 30 m)
- e2b	Thanétien moyen (III)	*Argile* de Vaux-sous-Laon (1 à 2 m)
- c4-6	Sénonien (II)	Craie blanche sans silex

- La couche e3 n'est présente que dans la partie sud de la carte ; la partie nord présente donc une lacune stratigraphique.
- Le calcul de pendages basé sur la supposition que les couches sont planes est cohérent dans la partie sud de la carte, mais non dans la partie nord. La structure géologique n'est donc pas tabulaire sur l'ensemble du terrain étudié.

On supposera donc par la suite que la structure est globalement monoclinale, les pendages de couches restant relativement faibles.

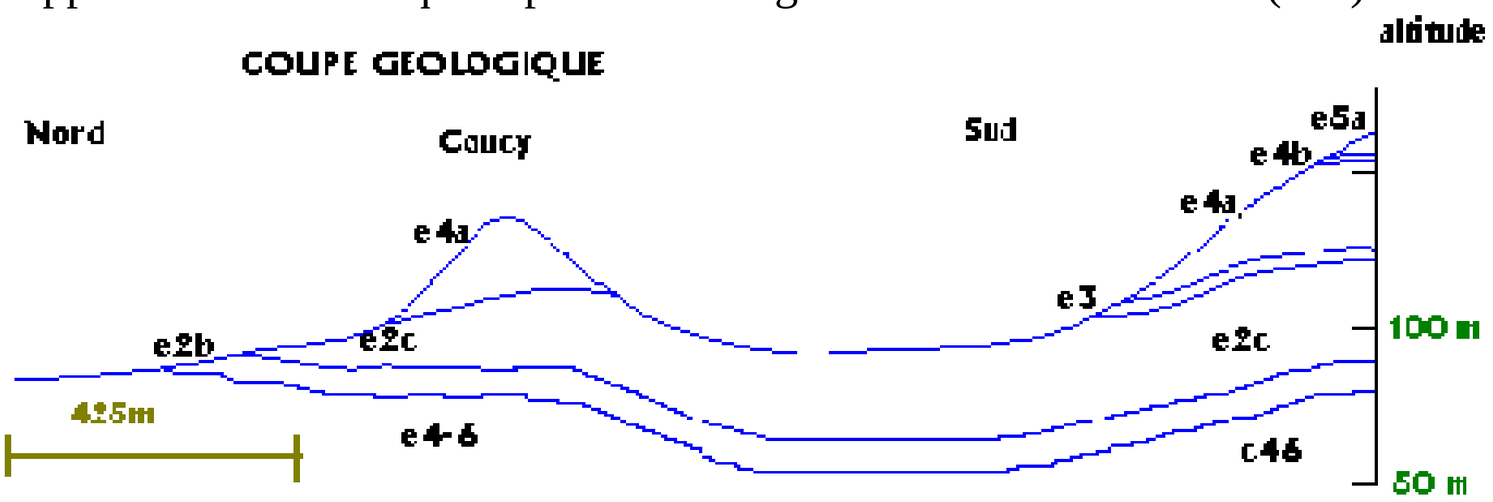
2) Modélisation numérique de la structure géologique du terrain.

Etant donné le caractère non strictement tabulaire de cette structure, elle ne pourra être l'objet d'une approximation linéaire ni polynomiale de degré 2.

Les altitudes des points des limites d'affleurement n'étant pas suffisantes pour approcher de façon satisfaisante, le recours à un programme Matlab réalisant des calculs de pendages puis les coupes géologiques associées est nécessaire.

L'approximation par des polynômes de degré 3 des limites de couches connaissant leurs pendages apparents, les coordonnées des points à l'affleurement et en supposant l'épaisseur des couches constantes donne par exemple le résultat suivant :

- la nappe du Thanétien qui repose sur l'argile de Vaux-sous-Laon (e2b). Elle alimente



notamment la source de la Croix-de-bois, située à l'ouest de Coucy (cf carte topographique)

- la nappe de la Craie qui repose sur l'argile de Laon (e4b). Cette nappe est très étendue dans la région et se trouve souvent superposée à d'autres nappes (référence **(7)**).
- Une nappe phréatique relativement profonde au sein de la craie du Sénonien utilisée par des captages et qui ne sera pas étudiée.

La structure géologique est ensuite numériquement modélisée en trois dimensions de la même façon que la topographie.

La connaissance de la nature des roches et de leurs caractéristiques pourra ensuite permettre des déductions quant au fonctionnement des nappes phréatiques.

II) Évaluation de la pollution de cette nappe

A) Prélèvement de l'eau à analyser et paramètres à envisager

Pour éviter toute modification, en particulier par des micro-organismes, les prélèvements d'eau réalisés dans des flacons de verre stériles, ont été dosés le plus rapidement possible (environ 12 heures) après avoir subit une trempe (environ 5°C) à l'abri de la lumière.
Voici la liste des eaux prélevées :

Origine de l'eau :	**Mode de prélèvement :**
Eau de la nappe	- Dans un puits chez un particulier (profondeur : 2m) - Dans un puits sur une place publique (profondeur : 1,5 m).
Eau d'une source (site de la Croix-de-bois)	Direct – eau stagnante
Eau du fossé du « chemin de la Sonnette »	Direct – très faible débit par temps sec

NB :
Le fossé du « chemin de la Sonnette » est à la sortie du village, sur les terrains de craie du Sénonien, qui passe au site de « La pelle au four », lieu d'implantation de la future zone de traitement des eaux usées. (cf carte topographique)
Ce fossé se situe donc en marge de la nappe des argiles de Vaux-sous-Laon à laquelle nous nous intéressons.
La zone étudiée étant très peu cultivée, on considère que la principale source de pollution des eaux est l'urée. Le rejet moyen d'urée par personne est de 400 mmol par jour.

Paramètre	**Prise en compte**	**Explications**
Paramètres azotés (d'après le contact **(4)**)		
Ions nitrates NO_3^-	Oui	Il s'agit d'une forme d'azote oxydé. Ces ions sont produits par minéralisation (oxydation par des bactéries ; $\Delta G_0' < 0$; spontané) de matières organiques azotées (protéines). Ils sont lessivables. (demi-équation électronique : $\mathbf{NH_4^+ + 3H_2O}$ « $\mathbf{NO_3^- + 8e^- + 10H^+}$; rapide. Précisons qu'il n'y a pas de minéralisation en profondeur.)
Ions ammonium NH_4^+	Oui	Il s'agit d'une forme d'azote (ammoniacal) réduit. Ces ions sont produits lors des premières transformations de l'urée (**NH2—CO—NH2**) par des bactéries possédant une enzyme (l'uréase) réalisant : $\mathbf{urée + 2H_2O \rightarrow CO_3^{2-} + NH_4^+}$. (constante de 0.36 j^{-1} d'après Tillotson et Wagenet)
Autres formes d'azote	Non	On ne prendra en compte que les formes d'azotes précédentes. Les ions nitrites (NO_2^-) par exemple, intermédiaires entre NH_4^+ et NO_3^- seront négligés. Précisons que la norme en ions nitrites est de 0,1mg/L
Autres paramètres (d'après le contact **(4)** et la référence **(6)**)		
MES	Non	Matières En Suspension
DBO_5	Non	Demande Biochimique d'O_2 en 5 jours
DCO	Non	Demande Chimique en O_2
Ions phosphates PO_4^{3-}	Non	Ces ions proviennent essentiellement des lessives et ne polluent que les eaux superficielles.
Autres ions ; métaux	Non	Négligés car le milieu est rural. Les sources de ces polluants (de type métaux lourds) sont considérées comme nulles.

B) Techniques de dosage des facteurs pris en compte

A la méthode de l'azote Kjeldahl (référence **(3)**) qui permet certes de déterminer la teneur en azote total (ammoniacal et organique) d'une solution, mais d'utilisation trop dangereuse sera préférée la méthode de Nessler (référence **(4)**) qui permet un dosage spectrophotométrique des ions NH_4^+. Son protocole sera explicité ci-après.
Enfin, le dosage des ions NO_3^-, possible par une méthode d'oxydo-réduction peu fiable, sera fait en utilisant des tests aquariophiles, peu précis mais d'utilisation simple et rapide.

C) Dosage des ions ammonium (NH_4^+) et des ions nitrates (NO_3^-)

Principe du dosage des ions ammoniums : Le *réactif de Nessler* ($KIHgCl_2$) en présence d'ions tartrate, de soude concentrée (à 9mol/L) et d'ions NH_4^+ libère les ions *iodure* qui donnent une coloration brune-orangé à la solution.

Le maximum d'absorption (ou densité optique) est obtenu pour λ=420 nm. On travaillera donc à cette longueur d'onde.

D'après la *loi de Beer-Lambert* : $A = \varepsilon(\lambda)*1*c_0$ où

A est l'absorbance, sans dimension

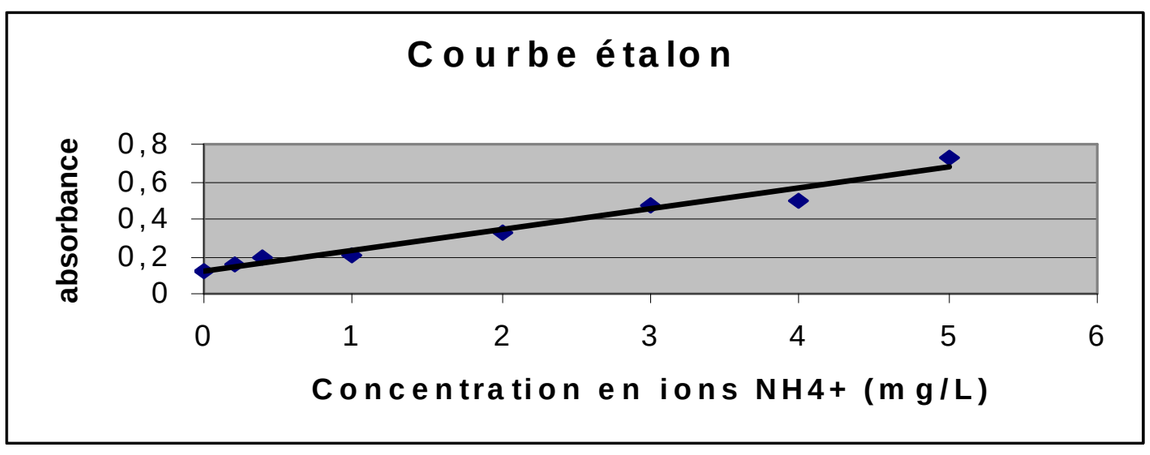
ε (λ) est le coefficient d'extinction molaire, exprimé en L/mg/cm

l est la longueur de la cuve, exprimée en cm

c_0 est la concentration en NH_4^+, exprimée en mg/L

L'absorbance est donc proportionnelle à la concentration en NH_4^+.

Protocole : On réalise une courbe étalon, représentée ci-dessous.



De cette droite on déduit la relation :

$$A = 0{,}114 * [NH_4^+] + 0{,}122$$

L'écart moyen entre valeurs calculées et valeurs mesurées est de :

$$0{,}025 = \frac{\Sigma\,|\text{valeur calculée} - \text{valeur mesurée}|}{\text{nombre de mesures}}$$

ce qui est faible, donc la méthode est précise.

Il suffit de mesurer l'absorbance d'une solution de concentration inconnue réalisée dans les mêmes conditions que l'étalon pour pouvoir, grâce à cette relation ou à la courbe, déduire la concentration recherchée.

Résultats des dosages réalisés :

Site de prélèvement	Absorbance	$[NH_4^+]$ (mg/L)
Source de la Croix-de-Bois	0,188	0,58
Puit dans le village (profondeur : 2m)	0,148	0,23
Fossé du « chemin de la Sonnette »	0,137	0,10

Conclusion : La norme se situe à ***0,5 mg/L***, la pollution en NH_4^+ est donc effective au niveau de la source. Elle est par contre inexistente dans les puits et dans le fossé où s'écoulent les eaux usées.

Protocole du dosage des nitrates : Des tests aquariophiles (Aquascience – Aquatest Nitrates NO_3) permettent de doser les nitrates. Après ajout des deux réactifs, mélange et attente le temps de la révélation, on peut évaluer un encadrement de la concentration recherchée par comparaison de la couleur obtenue avec une échelle de couleurs (de jaune à rouge).

Critique : Cette méthode est très approximative. Elle ne permet pas un titrage précis mais un encadrement : de 0 à 20, de 20 à 50, de 50 à 100 et de 100 à 200 mg/L. Néanmoins, elle s'avère rapide et peut être utilisée sur le terrain.

Résultat des dosages réalisés :

Site de prélèvement	$[NO_3^-]$ (mg/L)
Source de la Croix-de-Bois	20-50
Puit dans le village (profondeur : 2m)	50 à 100
Fossé du « chemin de la Sonnette »	Presque nulle

D) Bilan quant à l'impact anthropique

La norme en matière de concentration en nitrates se situe à ***50 mg/L***. La pollution en NO_3^- est donc effective au niveau de la nappe. Elle est presque nulle dans le fossé où s'écoulent les eaux usées, sans doute suite à une dilution et à l'utilisation des formes d'azote NO_3^- et NH_4^+ par des micro-organismes et végétaux.

Si la pollution des eaux est effective, elle reste peu importante, ce qui est relativement inattendu étant donné l'absence de tout-à-l'égout.

Une hypothèse pouvant expliquer ce fait est que, lors de l'infiltration des eaux dans le sol, les sables ou autres roches auraient pu retenir certains polluants, ou tout simplement qu'il existe des zones d'accumulations liées à la géométrie du mur de la nappe.
Cependant, de telles hypothèses sont difficilement vérifiables.

L'étude de ces éventuelles zones de rétentions nécessite donc une modélisation numérique de la nappe phréatique tenant compte des apports en urée. Les réactions rapides pourront être considérées comme totales et instantanées selon l'échelle de temps adoptée.

III) Évolution naturelle de la répartition des ions nitrates au sein de la nappe

A) Étude de l'écoulement de la nappe.

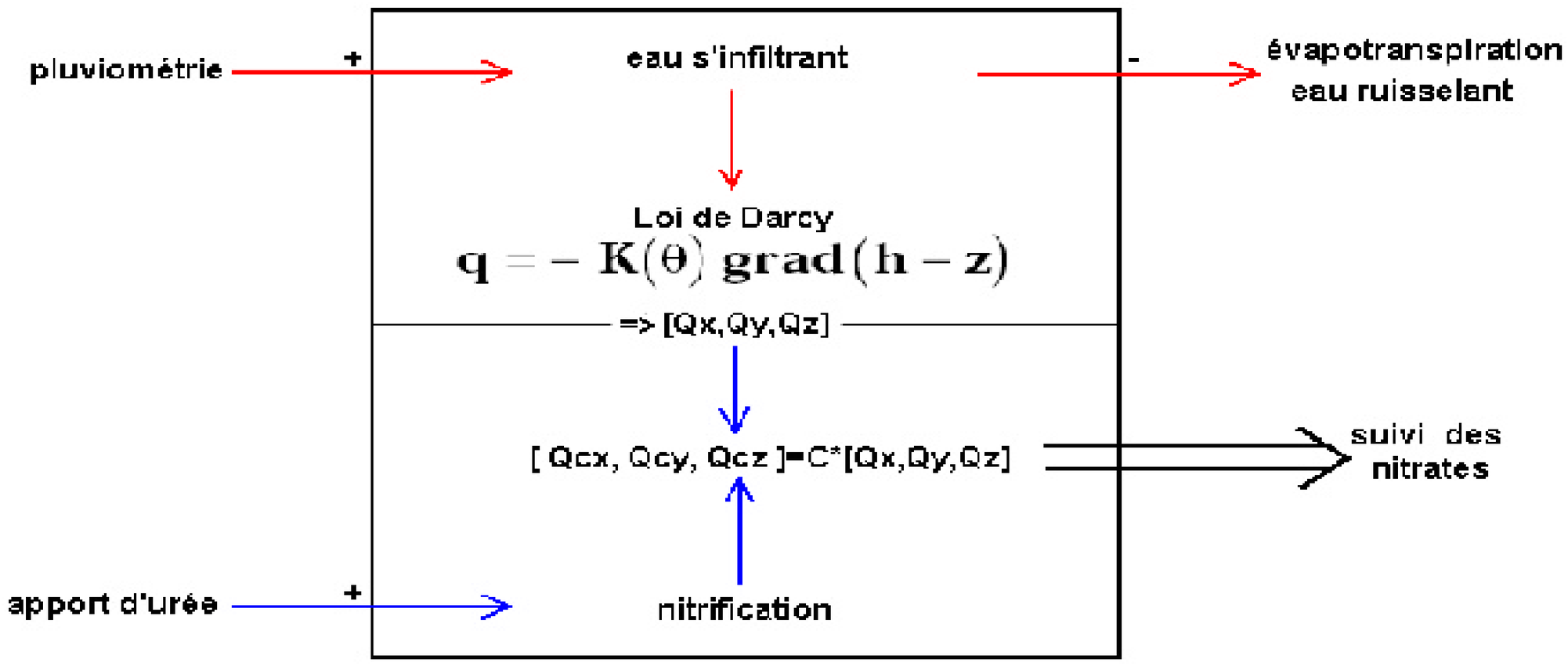
L'écoulement sera étudié grâce aux relevés pluviométriques des années 2001 et 2002 et une description hydrodynamique de la nappe phréatique.
Les résultats de l'étude de faisabilité géotechnique des travaux d'implantation des installations d'assainissement, permettent d'approcher certaines caractéristiques des sols étudiés, telles que les porosités totales, efficaces, les perméabilités au niveau des forages etc.
La modélisation nécessite la connaissance de la quantité d'eau s'infiltrant, aussi est-il nécessaire de tenir compte de l'évapotranspiration et du ruissellement.
Les valeurs choisies pour ces paramètres sont des valeurs moyennes fournies par la littérature, de même que les perméabilités en grand des couches non concernées par l'étude géotechnique.
L'écoulement est modélisé en deux temps : au sein d'un même schéma itératif ont lieu le calcul de la quantité d'eau s'infiltrant en tenant compte des phénomènes de ruissellement, et le calcul des déplacements de masses d'eau au sein des roches.
La charge h fonction de la masse volumique des roches varie avec leur taux d'hydratation (θ), ainsi que la perméabilité $K(\theta)$ exprimée en mètres carrés.
Deux modèles décrivant la variation de K en fonction de θ sont aisément applicables dans le cadre de la modélisation :

- le modèle de Van Genuchten et Nielsen (1985), soit :

$$\frac{K}{K_s} = \sqrt{S_e}\left[1-\left(1-S_e^{1/m}\right)^m\right]^2 \quad \text{avec} \quad S_e = \frac{\theta-\theta_r}{\theta_s-\theta_r}$$

- le modèle de Campbell (1985), soit : $\frac{K}{K_s} = \left(\frac{\theta}{\theta_s}\right)^m$

La constante m étant égale à 1 - (1 / n) avec n rendant compte de la texture du sol.
Le schéma d'un cycle de calcul est le suivant.

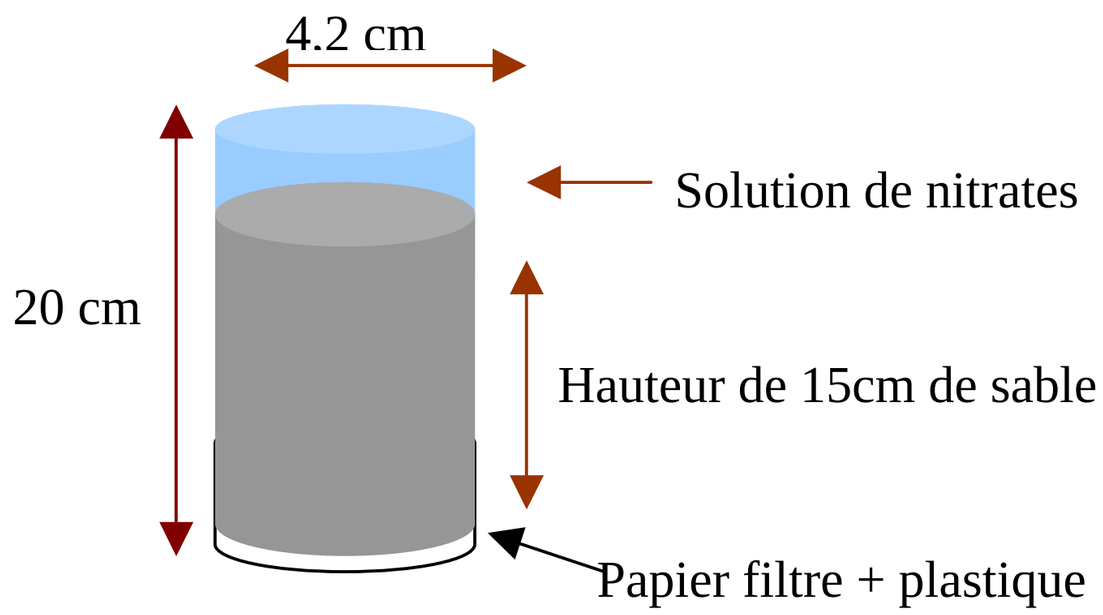


B) Étude des transports des ions étudiés au sein de la nappe.

1) Mise en évidence et quantification d'une rétention.

Des échantillons de sables ont été prélevés à l'affleurement, dans une ancienne carrière. Ces sables sont issus des couches traversées par les eaux .

Protocole pour la rétention des ions nitrates :



Dans six morceaux de tubes on introduit du sable de sorte qu'il soit au même degré de compaction que dans son site naturel ainsi que des solutions contenant des ions nitrates.

Au bout de cinq jours on récupère le filtrat en perçant par le bas du tuyau.

Les solutions introduites sont :
Tubes 1 à 3 : 30 mL d'eau distillée (expérience témoin)
Tubes 4 à 6 : 30 mL d'une solution à 100 mg/L de nitrates

Résultats :
Tubes 1 à 3 : entre 0 et 10 mg/L
Tubes 4 à 6 : entre 50 et 70 mg/L

Conclusion : Il y a rétention d'ions nitrates par les sables testés.

La détection de nitrates dans les échantillons témoins pourrait être due à un manque de fiabilité des tests nitrates utilisés ou à un mauvais lavage des sables testés.
Cependant, la faible précision des mesures faites ne permet pas de quantifier la rétention toutefois mise en évidence grâce à la répétition de l'expérience.

Cas des ions ammoniums :

Le même type d'expérience a été tenté pour ces ions, avec des concentrations différentes (0 ; 0,2 ; 0,4 ; 0,6 ; 0,8 ; 1 ; 2 mg/L) mais un problème s'est posé. En effet, les solutions récupérées n'étaient pas parfaitement limpides (malgré les tentatives de centrifugation) vraisemblablement à cause de la présence de fines particules d'argile. Les dosages par spectrophotométrie se sont avérés impossibles (nécessité de solutions parfaitement limpides) et les quelques essais ont donné des résultats inexploitables.
Nous nous en remettrons par la suite au fait, énoncé dans la littérature, que les ions ammoniums ne sont transportés que sur de très courtes distances.
Ce transport sera donc négligé. L'impossibilité de quantifier cette rétention implique la nécessité de sa modélisation par voie numérique.

2) Caractérisation du transport des ions nitrates au sein de la nappe.

On définit le flux diffusif, convectif et dispersif, respectivement q_d, q_c et q_m de la façon suivante :

$$\mathbf{q_d = -D_d\theta\frac{\partial C_L}{\partial z}},\ \mathbf{q_c = q\,C_L},\ \mathbf{q_m = -D_m\theta\frac{\partial C_L}{\partial z}}$$

On fait l'hypothèse que la valeur du débit q de la nappe multipliée par la concentration en polluants est importante devant les débits q_d et q_m , hypothèse qui sera vérifiée à posteriori.
La constante de $0.33 mg.m^{-2}.j^{-1}$ pour les réactions de dénitrifications permet sur une échelle de temps au minimum journalière de négliger ces réactions au vu de l'importance des apports.
L'espèce étudiée sera donc l'ion nitrate.
Les phénomènes d'adsorption/désorption seront négligés.

C) Mise en place d'une modélisation numérique

1) Application de la loi de Darcy

Le terrain est à cette fin découpé en unités de volumes, des cubes de 4.24 mètres de côté.
La saisie des caractéristiques des couches permet l'obtention de matrices à trois dimensions et l'application directe de la loi de Darcy. La charge est calculée pour chaque unité de volume, et le gradient est obtenu par soustraction de la matrice charge à une matrice identique hormis

une translation d'une unité, respectivement selon l'axe des x, des y et des z orienté vers le bas.

Les matrices flux obtenues sont multipliées par la surface d'échange pour obtenir des débits volumiques.

Ces débits volumiques calculés à partir de la perméabilité à saturation des roches se verront modifiés à chaque cycle de calcul pour tenir compte de la variation de la teneur en eau.

Les débits obtenus sont particulièrement grands devant les coefficient de diffusion qui sont de l'ordre de 10^{-9} m²/s multipliés par la charge maximum envisageable, ce qui justifie l'hypothèse faite étant donné que le gradient de concentration ne peut être supérieur à la concentration elle même.

2) <u>Modélisation de l'écoulement.</u>

On crée une matrice V contenant les volumes d'eau en mètres cubes contenue par chaque unité de volume et on utilise les débits calculés pour décrire l'écoulement, en tenant compte du volume de saturation des roches, et de leur taux de rétention qui influe sur la quantité d'eau cédée. Les calculs impliquent une translation d'une unité du contenu des matrices débits, ce qui entraîne au niveau des limites du cadre d'étude la définition d'une zone tampon à volume d'eau fixé, évitant ainsi les effets de bord.

3) <u>Modélisation du transport des ions nitrates</u>.

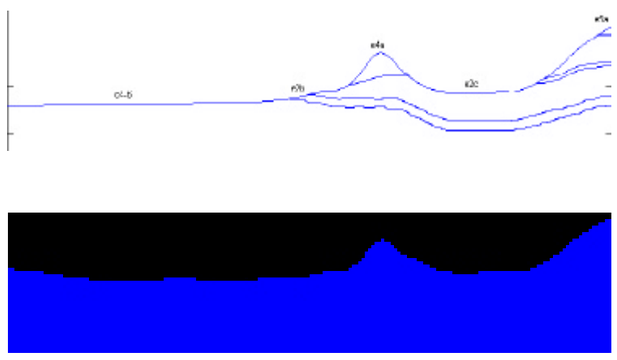
Le script réalisé lit une carte représentant le village de Coucy-les-Eppes sur laquelle chaque foyer est repéré par un pixel d'une couleur définie. On prendra par la suite un apport moyen journalier de 3 fois 400 mmol d'urée entièrement transformée en ammonium, puis en nitrite et en nitrate.

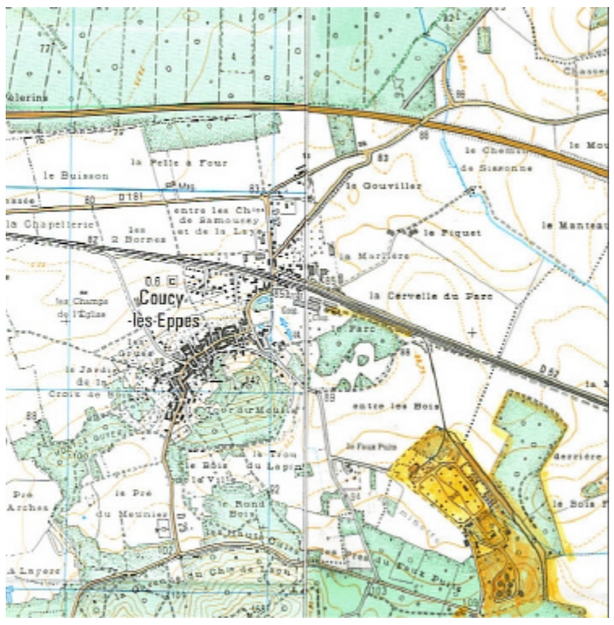
Que peut-on envisager pour répondre à cette pollution des sols conjecturée par la modélisation ?

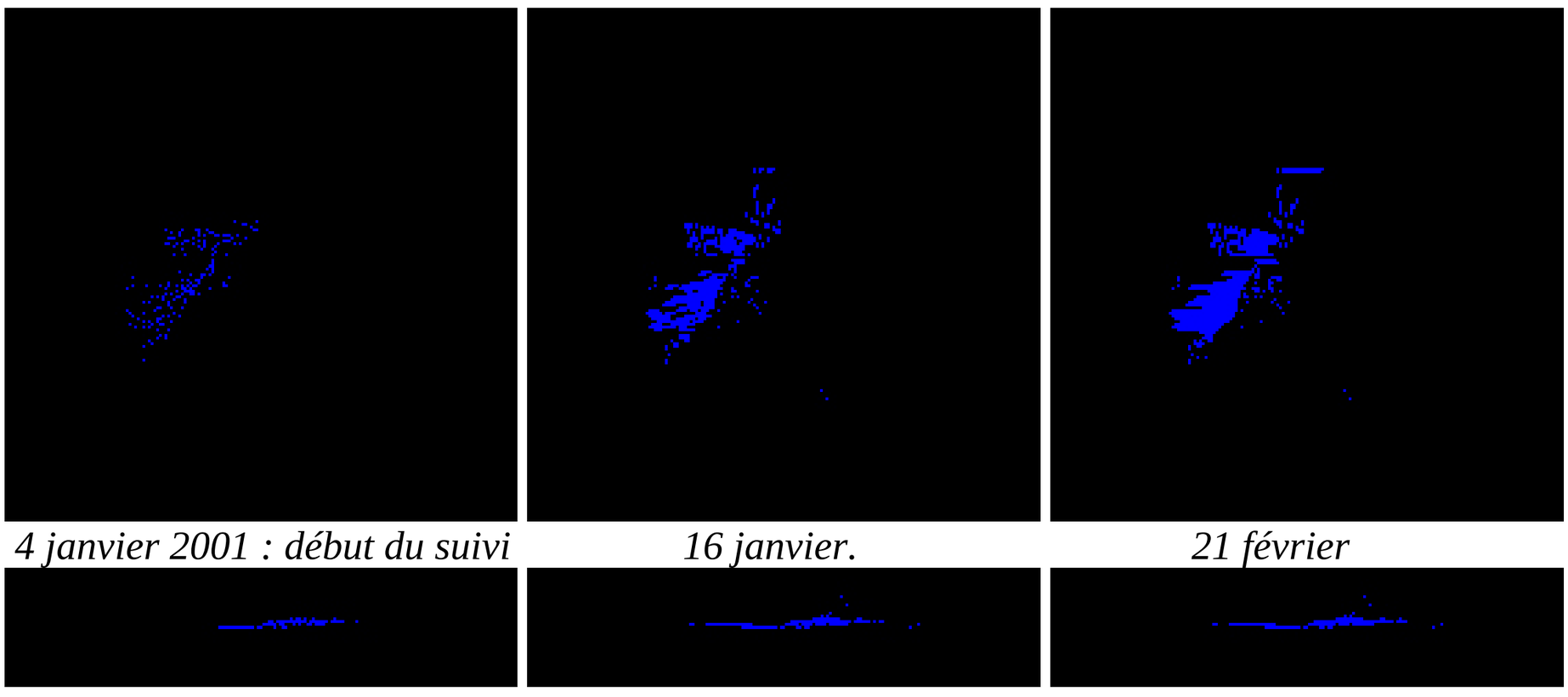
<u>IV) Évolution de cette situation compte tenu de l'installation d'un système d'assainissement</u>

<u>A) Validation du modèle et évolution des concentrations en polluants</u>.

Le suivi par la méthode précédemment évoquée de la répartition des concentrations en nitrates supérieures à la norme donne les résultats suivants, les premières vues étant des vues de haut, de même orientation que la carte topographique, et les deuxièmes vues des vues en coupe d'orientation Nord-Sud :





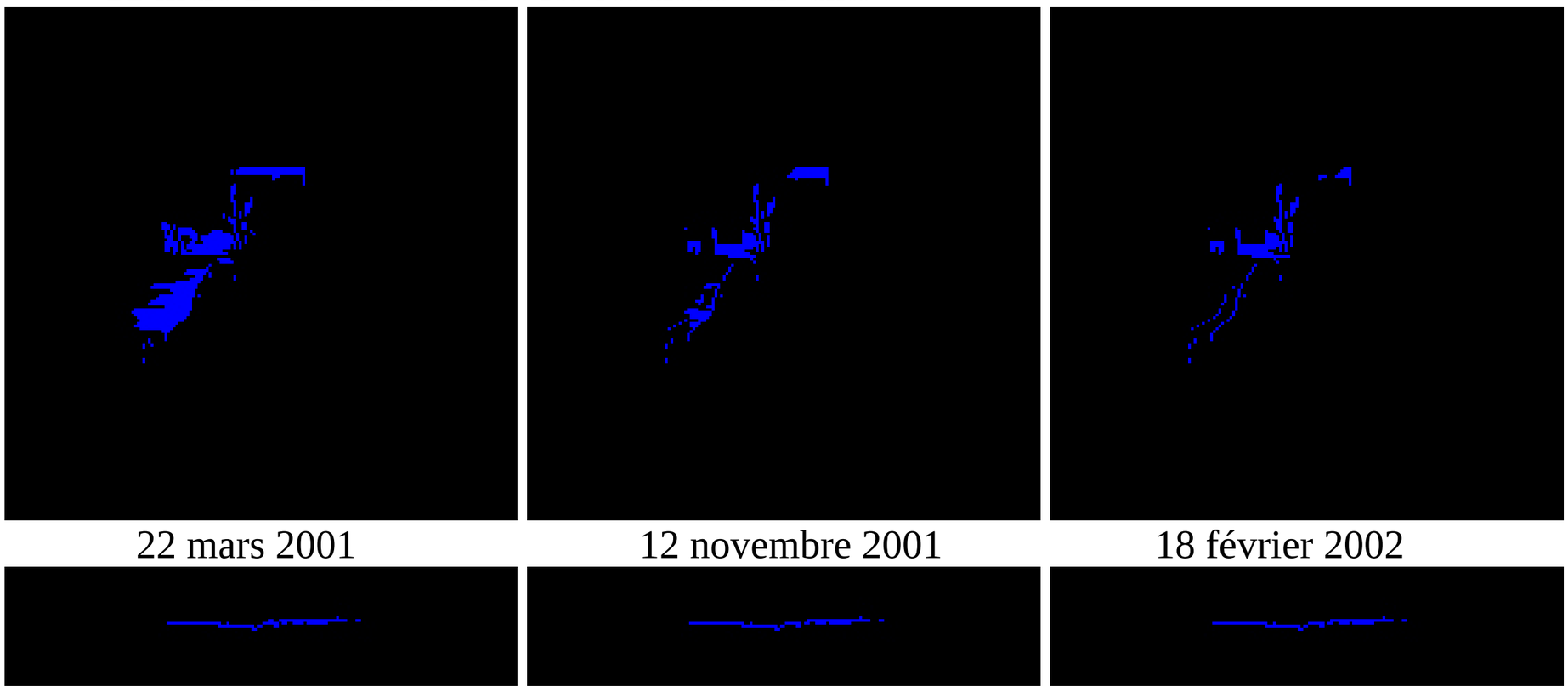


4 janvier 2001 : début du suivi *16 janvier.* *21 février*

On considère l'apparition à t=0 de l'apport d'urée. On constate l'établissement d'une répartition stable dès le 28 janvier, les concentrations en nitrates n'évoluant plus que de façon négligeable excepté localement. Cela relève d'un équilibre entre convection et lessivage des ions nitrates.
Le lessivage ne permet donc pas la dépollution des sols.
La concentration calculée à l'équilibre au niveau de la zone où est localisée le puit est de 43 $mg.L^{-1}$, ce qui coïncide avec les mesures faites.
Cependant, au niveau de la zone de la Croix-de-bois, elle est de 62 $mg.L^{-1}$, ce qui est conséquemment supérieur à la concentration mesurée.
Cette erreur peut être attribuée à la présence d'une végétation importante. En effet, l'influence de la végétation n'a pas été prise en compte et pourrait avoir contribué à la diminution de la concentration en nitrate.

B) Prévisions à long terme

L'installation du système d'assainissement permettra l'interruption des apports d'urée grâce à la mise en place d'un système d'épuration par lagunage des eaux usées. Mais quelle sera l'influence de cette interruption sur l'équilibre entre rétention et lessivage des nitrates précédemment mis en évidence ?
Pour répondre à cette question, l'apport d'urée a été interrompu au sein du modèle à compter du 11 mars 2001. On observe les résultats suivants :



Après 18 mois à compter du 18 février, on observe une diminution de 17% du niveau de pollution aux points de rétention observés initialement.
Ceux-ci correspondent à des zones qui verront leur concentration en nitrates varier peu et lentement.

C) Solutions envisageables et prévision compte tenu de l'installation de l'assainissement.

Au vu des résultats donnés par la modélisation numérique, si celle-ci se confirme à long terme, la dépollution naturelle des sols suite à l'interruption de l'apport d'urée sera d'autant plus lente que le flux de concentration par convexion est un produit d'un débit (qui restera probablement inchangé) par la valeur de la concentration qui tendra à décroître.
De plus, les points d'accumulations sont tous localisés au sein du village, dans des zones où la végétation est peu importante, et donc la non prise en compte de la consommation d'azote par les végétaux n'a vraisemblablement pas d'incidence sur ce résultat.
L'importance de ce lessivage tendrait donc à diminuer au fil du temps.
Un retour à une concentration normale en nitrate pourrait être accéléré par la plantation au niveau des zones atteintes de végétaux à grande consommation de nitrates et dont les racines s'enfoncent à une profondeur correspondant à celle des zones de rétention, en l'occurrence un intervalle de profondeur de 1,60 mètre à 16 mètres.

Conclusion

L'hypothèse d'une pollution des sols est vérifiée, et semble liée à une insuffisance du lessivage des ions nitrates. La mise en place d'un système d'épuration des eaux usées fonctionnant sur le principe du lagunage représentera dès Octobre 2004 le commencement de l'auto-épuration efficace des eaux de cette nappe phréatique et des sols concernés.
Cette auto-épuration peut être accélérée par la plantation, au niveau des zones d'accumulation des nitrates, d'arbres susceptibles de consommer les nitrates incriminés.
On peut noter toutefois que cette pollution reste localisée au sein d'une nappe phréatique de faible importance et non utilisée, et que les nitrates qui atteignent la nappe phréatique de la craie du Sénonien utilisée pour l'approvisionnement en eau potable sont suffisamment dilués pour représenter un apport négligeable.
Une épuration rapide de la nappe phréatique superficielle permettrait cependant de garantir une qualité suffisante pour sécuriser les quelques puits encore utilisés, par exemple pour abreuver des animaux.
En outre, le village compte un certain nombre de puits non déclarés, traversant parfois la couche argileuse et pouvant alors représenter une fuite des eaux de la nappe perchée vers celles de la nappe du Sénonien, avec le danger de fournir une eau potentiellement polluée quoique venant de la couche crayeuse profonde.

Principaux contacts

- **(1)** Monsieur M. Fontaine, *maire* de Coucy-les-Eppes
- **(2)** Monsieur J. Leroy, de la Direction Départementale de l'Equipement de Soissons, *responsable des travaux*, contacté par téléphone (au 03 23 75 85 74)
- **(3)** Monsieur H. Denudt, *hydrogéologue* agréé en matière d'hygiène publique, chargé de l'étude réalisée en 2002, contacté par téléphone (au 03 20 60 45 62)
- **(4)** Monsieur J.P. Pardoux, *ingénieur en environnement et en aménagements du territoire* de la Chambre d'Agriculture de la Somme, contacté par téléphone (au 03 22 33 69 28)
- **(5)** Météo France, centre départemental de l'Aisne (aérodrome de Saint Quentin Roupy), contacté par téléphone (au 03 23 50 81 81) puis par Internet (cdm02@meteo.fr).

Bibliographie

(1) Carte IGN n°27100 (Laon) et 28100 (Sissone) au 25 000ème , 1994

(2) Carte géologique obtenue sur : http://infoterre.brgm.fr/ sélection de l'icône : « Accès aux cartes et aux données » puis recherche par communes : « Eppes » puis choix de « Coucy-les-Eppes »

(3) Manipulations d'analyse biochimique, de la collection « Biosciences et Techniques », aux éditions Doin (1996), et rédigé par M. Gavrilovic, M.J. Maginot, C. Schwartz-Gavrilovic et J. Wallach [pages 52, 65]

(4) http://www.ac-poitiers.fr/svt/res_loc/hydro/para_eau/Para-nh4.htm ainsi que le Jean Rodier, éditions Dunod, huitième édition, chapitre 8-1 : Dosage de l'ammonium

(5) Rapport de la société NATURE et TECHNIQUE (S.I.N.T.) sur le traitement des eaux usées et l'assainissement grâce à des aménagements aquatiques, consulté à la mairie

(6) http://pst.chez.tiscali.fr/svtiufm/hydrogeo.htm#nappesetaquiferes

(7) copies de transparents de résumés de cours de physique des sols. école polytechniques fédérale de Lausanne. Rédigés par A. Mermoud.

(8) Complément de cours de physique des sols de l'école polytechniques fédérale de Lausanne rédigés par Michel Dysli. :
http://lmswww.epfl.ch/commun/pdf_doc_sols/Complements_cours_mec._sols.pdf

(9) A fast algorithm for the estimation of the equivalent hydraulic conductivity of heterogeneous media, WATER RESOURCES RESEARCH, volume. 36, numéro 12, pages 3567 à 3580, décembre 2000

(10) Comparaison entre différentes formulations de conductivité équivalente en milieu poreux non saturé , publication de *B. Belfort, F. Lehmann et Ph. Ackerer* .

(11) Cours de géophysique de l.Université de Lausanne . Principes de base - D. Chapellier

(12) Vers une modélisation hydrologique adaptée à l'évaluation des pollutions diffuses : prise en compte du réseau anthropique. Application au bassin versant de Naizin, thèse de Nadia Carluer, annexe 1 : http://www.lyon.cemagref.fr/doc/these/carluer/carluer7.pdf

Annexe 1 : détail des scripts matlab.

%script_topographie ; lecture de la carte topographique.

```
texte=input('entrez le chemin du fichier à lire ','s');
texte=['I=imread(' texte ');'];
eval(texte);

H=[]; %on initialise la matrice destinée à contenir les coordonnées des pixels apparetenant aux lignes de niveau.
taille=size(I);
h = waitbar(0,'interprétation des couleurs');
%boucle permettant de tester la couleur de chaque pixel de l'image à
%traiter et de la comparer à l'échelle de couleur établie a postériori.
echellemin=input('entrez les valeurs minimum en [rouge,vert,bleu] sous cette forme ');
echellemax=input('entrez les valeurs maximum en [rouge,vert,bleu] sous cette forme ');
for i=1:taille(1)

G=find(((I(i,:,1)>=echellemin(1))&(I(i,:,1)<=echellemax(1)))&((I(i,:,2)>=echellemin(2))&(I(i,:,2)<=echelle
max(2)))&((I(i,:,3)>=echellemin(3))&(I(i,:,3)<=echellemax(3)))); %on cherche les coordonnées des points
donc la couleur correspond à l'encadrement donné.
    u=zeros(1,length(G))+i;
    if ~isempty(u)
    H=[H [u;G]]; %on concatène les coordonnées de ces points à H.
    end
end
close(h)
visualiseH(H,max(taille)) %appel d'une fonction qui permet de visualiser les données contenues dans H et
de juger de la validité de l'encadrement de la couleur des lignes de niveau.
%appel de la fonction regroupement qui permet de ranger les coordonnées
%des points appartenants aux lignes de niveau contenus dans H(1:2,:)
%par appartenance aux différentes courbes de niveau.
save H.mat H
G=regroupement(H); %la fonction regroupement renvoie une chaine de caractère à autant de lignes qu'il y a
de lignes de niveau.
save G.mat G

taille=size(I);
taille3=size(G);
disp(['il y a ' num2str(length(H)) ' points à considérer'])
H=[H;[zeros(length(H))-20000]];  %on concatène verticalement un vecteur de meme longueur que H ne
contenant que des valeurs d'altitude impossibles, ce qui permet de créer une troisième ligne qui contiendra
les altitudes associées à chaque point.
g = waitbar(0,'saisie des altitudes');
for i=1:taille3(1)
    waitbar(i/taille3(1),g)
    Iseconde=I;
    matrice=eval(eval(['G(' num2str(i) ',:)'])); %le contenu de G doit etre converti en vecteur pour etre utilisé.
    h = waitbar(0,'dessin de la courbe de niveau');
    for k=1:length(matrice)
        f=matrice(k);
        waitbar(k/length(matrice),h)
        Iseconde(H(1,f),H(2,f),:)=[255 0 255]; %on colore en rose la ligne de niveau dont l'utilisateur devra
entrer l'altitude pour la rendre reconnaissable.
    end
close(h)
```

%il est nécessaire de réaliser un affichage de l'image initiale I et de
%l'image qui visualise uniquement la ligne de niveau étudiée ; Iseconde.
image(Iseconde), axis image
    altitude=input(['entrez l''altitude de la courbe de niveau dessinée en rose.']); %saisie de l'altitude de chaque ligne de niveau.

    pas=input('entrez le pas avec lequel l''altitude sera entrée '); %si le nombre de points a considérer est trop grand, il est nécessaire de ne tenir compte que de points régulièrements espacés.
    for j=1:pas:length(matrice)
        H(3,f)=altitude;
    end
end
r=find(H(3,:)==-20000); H(:,r)=[]; %on supprime les points pour laquelle aucune altitude n'a été saisie.
save Hfinal H %on obtient une matrice contenant en ligne 1 et 2 des coordonnées de noeuds, et en ligne 3 les altitudes des noeuds.

%fonction regroupement
function G=regroupement(H)
%regroupement permet de dégager d'une matrice contenant des coordonnées de
%points sous la forme de vecteurs colonnes une chaine de caractère dont
%chaque ligne représente les indices colonnes dans la matrice de départ
%correspondant au points d'un meme ensemble géographique (ligne de niveau,
%limite de couche...
taille=size(H);
G=''; %toutes les lignes de niveau ne comportant pas le meme nombre de points, la concaténation ne sera possible que sous la forme d'une chaine de caractère.
F2=[1:taille(2)]; %nous n'allons pas travailler sur les vecteurs colonnes de H représentant les coordonnées des points mais les indices de ces colonnes dans la matrice H qui reste inchangée.
y=length(F2);
d = waitbar(0,'extraction des nuages de points');
while (length(F2)~=0) %nous allons par la suite etre ammenés à déléter du vecteur F2 les indices corespondant aux points traités jusqu'à ce que tout les points aient été traités. De plus la variable compteur ne doit pas dépasser la longueur de H.
    i=F2(1);F2(1)=[];M=[i]; %on prend un indice correspondant à un point de référence que l'on supprime de F2 pour qu'il ne soit pris en compte qu'une seule fois, et qu'on concatène injecte dans M qui contiendra les indices des points de meme ligne de niveau que le point de référence.
        S=find((sqrt(abs(H(1,F2)-H(1,i)).^2+abs(H(2,F2)-H(2,i)).^2)<=3)); %on repère les indices des éléments situés à moins d'une distance de 3 pixels du pixel de référence.
        waitbar((y-length(F2)+1)/y,d);
        M=[M F2(S)];
        F2(S)=[];
    t=length(F2);
    K=[-3 -4]; %on prend un vecteur K quelconque de longueur 2 et à membres consécutifs non égaux.
    r=length(M)-1;
    while K(length(K))~=K(length(K)-1)%la boucle est effectuée tant que la valeur du dernier élément de K est différente de celle de l'avant-avant dernier élément de K.
        D=length(M);
        for k=D-r:D   %on parcours l'ensemble des indices contenus dans M exepté celui du point de référence.
            L=M(k); %la variable l contient successivement chaque indice de M à l'exeption de celui du point de référence.
            t=length(F2);

```
        S=find((sqrt(abs(H(1,F2)-H(1,L)).^2+abs(H(2,F2)-H(2,L)).^2)<=3)); %on repère les indices des éléments situés à moins d'une distance de 3 pixels du pixel de référence.
        r=length(S);
        waitbar((y-length(F2)+1)/y,d);
        M=[M F2(S)]; % on injecte dans M ces indices.
        F2(S)=[];  %on supprime les indices de ces points pour éviter qu'ils ne soient repris en compte.
        t=length(F2);  %la variable t est remise à jour suite à l'éventuelle délétion.
        D=length(M);
    end
    K=[K t];%on injecte la longueur du vecteur F2 dans le vecteur K, ainsi le test est répété tant que F2 contient des indices de points a proximité directe de points de M
  end
  G=strvcat(G,mat2str(M));D=length(M) %on concatène verticalement le vecteur M, une fois complet, sous forme de chaine de caractère à G
    save G.mat G M K F2 r %une sauvegarde automatique des variables importantes permet de continuer l'opération après une éventuelle interruption
end
close(d)
%chaque ligne de G correspond finalement à la transformation en chaine de
%caractères d'une matrice contenant les indices colonnes dans H de toutes
%les coordonnées des points d'une meme ligne de niveau.
```

%calcul_pendage :

```
load donnees_site2.mat
for k=2:8
%    altitude=[];
%    H=eval(['Couche' num2str(k) 'lim']);
%    for i=1:length(H)
%        altitude=[altitude Z(H(1,i),H(2,i))];
%    end
%    eval(['Couche' num2str(k) 'lim=[Couche' num2str(k) 'lim;altitude];'])
%end
%calcul du pendage d'une couche.
H2=eval(['Couche' num2str(k) 'lim']);
H=eval(['Couche' num2str(k-1) 'lim']);

%if k=7
%   H3=Couche3lim;
%end
%On cherche l'épaisseur apparente minimale pour s'en servir d'étalon.
donnee1=[];
for i=1:length(H)-1
    a=H(:,i);
    dist=sqrt((H2(1,:)-a(1)).^2+(H2(2,:)-a(2)).^2+(H2(3,:)-a(3)).^2);
    epaisseur_apparente=min(dist);
    f=find(dist==min(dist));
    b=H2(:,f);
    y=abs(a(3)-b(3));
    donnee1=[donnee1 epaisseur_apparente];
end
r=find(donnee1>10*min(donnee1));
donnee1(r)=[];
donnee=mean(donnee1);
```

```
coord=[];e=[];diff=[];donnees=[];
for i=1:2:length(H)-1
    %premier point :
    a=H(:,i);
    dist=sqrt((H2(1,:)-a(1)).^2+(H2(2,:)-a(2)).^2+(H2(3,:)-a(3)).^2);
    epaisseur_apparente=min(dist);
    f=find(dist==min(dist));
    b=H2(:,f);
    y=abs(a(3)-b(3));

    %deuxième point
    a2=H(:,i+1);
    dist2=sqrt((H2(1,:)-a2(1)).^2+(H2(2,:)-a2(2)).^2+(H2(3,:)-a2(3)).^2);
    epaisseur_apparente2=min(dist2);
    f2=find(dist2==min(dist2));
    b2=H2(:,f2);
    y2=abs(a2(3)-b2(3));
    if
((donnee(1,1)+donnee(1,1)*(25/100))>epaisseur_apparente)&((donnee(1,1)+donnee(1,1)*(25/100))>epaisse
ur_apparente2)
        donnees=[donnees;[epaisseur_apparente,y,i,f];[epaisseur_apparente2,y2,i+1,f2]];
        coord=[coord [a(1);a(2);a(3)]];e=[e epaisseur_apparente];diff=[diff y];
    end
end
taille2=size(donnees);epaisseurs_apparentes=[];ys=[];
for i=1:2:taille2(1)-1
    epaisseurs_apparentes=[epaisseurs_apparentes (donnees(i,1)-donnees(i+1,1))];
    ys=[ys (donnees(i+1,2)-donnees(i,2))];
end
pendage=abs(atan(ys./epaisseurs_apparentes));
epaisseur=abs(e.*sin(pendage)+diff.*cos(pendage));
g=find(epaisseur<(mean(epaisseur)/4));
epaisseur(g)=[];pendage(g)=[];coord(:,g)=[];
nv_coord=coord;
e_vert=(epaisseur./abs(cos(pendage)));
g=find(e_vert>(mean(e_vert)+mean(e_vert)*(25/100)));
nv_coord(3,:)=nv_coord(3,:)-abs(e_vert);
nv_coord(:,g)=[];
eval(['Couche' num2str(k) 'lim=[Couche' num2str(k) 'lim nv_coord];'])
eval(['Couche' num2str(k) 'data=[pendage;epaisseur];'])
end
```

%écoulement :

```
% modélisation de l'écoulement.
%donnees : Qx,Qy,Qz,Z, bZ0, K
taille=size(Qx);
pluvio=([0.6 0 0.4 7.4 0.2 2.6 0 3.2 0.2 0.2 0.2 0 1.6 0.6 0.2 0 0 0 0 0.2 1.6 4.6 0.8 0 0.2 0 0 6.6 1.8 0 0.2 0.2
3.4 0 0 0 1.6 5.2 11.2 0.2 7.4 0.2 6.2 5.4 3.8 5.2 3.0 3.2 3.4 0.2 0.2 0.4 0 0.8 2.0 1.6 0.2 0 5.8 3.2 5.6 0 14.6 0
0 0.2 1.2 1.8 0 0 0 0 0 3.8 0 0 0 0.6 0 0 0.2 0 0 2.4 2.2 2.0 3.0 0 .0 14.2 9.2 0.4 0 1.6 5.6 1.8 0 0 0.2 0 0 0.6
0.6 3.2 4.8 1 2 8.2 0 0.2 1.8 6.4 12 3.8 0.2 0 0.4 7.6 13.2 0.6 0 0 0 0 0.2 0 2.8 0.6 0 0 12.6 59.2 0.4 0.2 0.2 0
0.6 1.4 1.8 3.8 0.2 5.2 0 1 1 5.2 1 5.4 0.2 6 11 0 0.2 0.2 21.8 3 14 2.2 0.2 0 0 0 0 0.2 0 0 0 0 0 0 0 0 1.6 0 0.4
0.4 0 0 0 0 0 1 0 44 0 9.2 0.8 0.2 13.4 5.8 8.2 14.8 0.6 0 6.6 7.2 6. 2.6 0.2 0 0 0 0 0.2 21.2 7.8 2.4 10.6 0.2
1.2 0 0 5 1.6 15.4 5.4 0.2 0 0 2 0.2 0 0.2 0 1.4 0 0 0.2 0 0.2 2.2 0 0.2 0 0 0 0 3.6 0.2 0.2 0 0 0.4 9.2 14 0.6 0
```

```
1.4 10 0 0.6 0 0.2 0.2 0 0 0.2 5. 1.8 0.2 0 4.2 0.2 0 0.2 0 0.4 0 0 1 5.8 4.6 0.2 0.2 3.6 0 0.2 0.2 0.2 3 11.8 15.6
10.6 0.2 6.2 0.2 0.2 7.6 12.6 0.2 15.4 0.2 0.2 0 0 0.4 0 1.6 0.4 0.2 0.2 0.2 6.4 5.8 2 2.2 0.4 0.2 9 3.8 12.2 1 0
0 0 11.6 0.6 2.2 0 8.2 0.2 1.2 5.2 0 .6 1.8 3.2 0.8 22 0 4. 0 0.6 0 5 4.2 0 1.6 1 3.8 3.2 0.2 3.8 3.6 2.4 3.8 5. 9.4
0.8 1.4 1 2.8 0 0 0 0 0 0 9.6 3 2.4 0.6 0.2 0.2 9.6]/1000)*echelle^2;
echelletemps=input('entrez l''échelle de temps ');
date=input('entrez la date sous la forme [jour mois] ');
modeleK=input('entrez 0 pour le modèle de perméabilité à non saturation de Van Genuchten, 1 pour celui de
Campbell et Brooks. ');
V=bZ0*pluvio(date(1)+1);%load Darcy.mat
C=V;C(:,:,:)=0;load x.mat
stop=0;
Qsx=Qx;Qsy=Qy;Qsz=Qz;
for i=1:100
    date(1)=date(1)+2;
    heau=V/(echelle^2);
P=g*1000*~bZ0.*heau;
Pxplus4=P;Pxplus4(1,:,:)=[];Pxplus4=cat(1,Pxplus4,P(taille(1),:,:));
Pyplus4=P;Pyplus4(:,1,:)=[];Pyplus4=cat(2,Pyplus4,P(:,taille(2),:));
Pzplus4=P;Pzplus4(:,:,taille(3))=[];Pzplus4=cat(3,P(:,:,1),Pzplus4);
dPsurdx=Pxplus4-P;dPsurdy=Pyplus4-P;dPsurdz=Pzplus4-P;
if modeleK==0
    KthetasurKs=sqrt(((V./(echelle^3))-(ne*(echelle^3)))./((n*(echelle^3))-(ne*(echelle^3)))).*(1-(1-
(((V./(echelle^3))-(ne*(echelle^3)))./((n*(echelle^3))-(ne*(echelle^3)))).^(n2./(n2-1))).^((n2-1)./n2)).^2;
else
    KthetasurKs=((V/(echelle^3))./((V~=0).*(n*(echelle^3))+(V==0).*1)).^((V~=0).*((1/(n2-
1))+3)+(V==0));
end
disp(max(max(max(V))))
disp(max(max(max(C))))
Qx=KthetasurKs.*Qsx-KthetasurKs.*K.*dPsurdx*echelle^2;
Qx=KthetasurKs.*Qsy-KthetasurKs.*K.*dPsurdy*echelle^2;
Qx=KthetasurKs.*Qsz-KthetasurKs.*K.*dPsurdz*echelle^2;
    disp(['itération ' num2str(i) ', pluviométrie à ce jour : ' num2str(pluvio(date(1))) ' '])
    V=V.*~bZ0+(pluvio(date(1))+pluvio(date(1)-1))*bZ0;
    for j=1:length(x)
        C(x(1,j),x(2,j),x(3,j))=C(x(1,j),x(2,j),x(3,j))+(((3*400*10^-3)/(3*1.5))/86800)*echelletemps;
    end
    Qcx=Qx.*C;Qcy=Qy.*C;Qcz=Qz.*C;
taille=size(Qx);
Qmoinsx=-Qx;
Qmoinsx(taille(1),:,:)=[];
Qmoinsx=cat(1,Qmoinsx(1,:,:),Qmoinsx);

Qcmoinsx=-Qcx;
Qcmoinsx(taille(1),:,:)=[];
Qcmoinsx=cat(1,Qcmoinsx(1,:,:),Qcmoinsx);

Qmoinsy=-Qy;
Qmoinsy(:,taille(2),:)=[];
Qmoinsy=cat(2,Qmoinsy(:,1,:),Qmoinsy);

Qcmoinsy=-Qcy;
Qcmoinsy(:,taille(2),:)=[];
Qcmoinsy=cat(2,Qcmoinsy(:,1,:),Qcmoinsy);
```

```
Qmoinsz=-Qz;
Qmoinsz(:,:,taille(3))=[];
Qmoinsz=cat(3,Qmoinsz(:,:,1),Qmoinsz);

Qcmoinsz=-Qcz;
Qcmoinsz(:,:,taille(3))=[];
Qcmoinsz=cat(3,Qcmoinsz(:,:,1),Qcmoinsz);

Qcmoinsx2=Qmoinsx.*C;Qcmoinsy2=Qmoinsy.*C;Qcmoinsz2=Qmoinsz.*C;


perte=((Qmoinsx>0).*Qmoinsx+(Qmoinsy>0).*Qmoinsy+(Qmoinsz>0).*Qmoinsz+(Qx>0).*Qx+(Qy>0).*
Qy+(Qz>0).*Qz)*echelletemps;
    rapport=abs((((max((V-(n-ne)*(echelle^3)),0)-perte)<0).*abs(max((V-(n-
ne)*(echelle^3)),0)./(perte.*(perte~=0)+(max((V-(n-ne)*(echelle^3)),0)~=0).*max((V-(n-
ne)*(echelle^3)),0).*(perte==0)+(max((V-(n-ne)*(echelle^3)),0)==0).*(perte==0)))+((max((V-(n-
ne)*(echelle^3)),0)-perte)>=0)));

pertec=((Qcmoinsx2>0).*Qcmoinsx2+(Qcmoinsy2>0).*Qcmoinsy2+(Qcmoinsz2>0).*Qcmoinsz2+(Qcx>0)
.*Qcx+(Qcy>0).*Qcy+(Qcz>0).*Qcz)*echelletemps;
rapportc=((C~=0).*abs(((C-
pertec)<0).*abs(C./(pertec.*(pertec~=0)+(C~=0).*C.*(pertec==0)+(C==0).*(pertec==0)))+((C-
pertec)>=0)));

rapportx=rapport;rapportx(taille(1),:,:)=[];rapportx=cat(1,rapportx(1,:,:),rapportx);
rapporty=rapport;rapporty(:,taille(2),:)=[];rapporty=cat(2,rapporty(:,1,:),rapporty);
rapportz=rapport;rapportz(:,:,taille(3))=[];rapportz=cat(3,rapportz(:,:,1),rapportz);
rapportx2=rapport;rapportx2(1,:,:)=[];rapportx2=cat(1,rapportx2,rapportx2(taille(1)-1,:,:));
rapporty2=rapport;rapporty2(:,1,:)=[];rapporty2=cat(2,rapporty2,rapporty2(:,taille(2)-1,:));
rapportz2=rapport;rapportz2(:,:,1)=[];rapportz2=cat(3,rapportz2,rapportz2(:,:,taille(3)-1));


gain=((Qmoinsx<0).*rapportx.*Qmoinsx+(Qmoinsy<0).*rapporty.*Qmoinsy+(Qmoinsz<0).*rapportz.*Qm
oinsz+(Qx<0).*rapportx2.*Qx+(Qy<0).*rapporty2.*Qy+(Qz<0).*rapportz2.*Qz)*echelletemps;
rapportgain=abs(((V-gain)>(n*echelle^3)).*((gain~=0).*(((n*echelle^3)-
V)./((gain)+(gain==0))+(gain==0)))+((V-gain)<=(n*echelle^3)));

rapportgainx=rapportgain;rapportgainx(taille(1),:,:)=[];rapportgainx=cat(1,rapportgainx(1,:,:),rapportgainx);
rapportgainy=rapportgain;rapportgainy(:,taille(2),:)=[];rapportgainy=cat(2,rapportgainy(:,1,:),rapportgainy);
rapportgainz=rapportgain;rapportgainz(:,:,taille(3))=[];rapportgainz=cat(3,rapportgainz(:,:,1),rapportgainz);
rapportgainx2=rapportgain;rapportgainx2(1,:,:)=[];rapportgainx2=cat(1,rapportgainx2,rapportgainx2(taille(1
)-1,:,:));
rapportgainy2=rapportgain;rapportgainy2(:,1,:)=[];rapportgainy2=cat(2,rapportgainy2,rapportgainy2(:,taille(
2)-1,:));
rapportgainz2=rapportgain;rapportgainz2(:,:,1)=[];rapportgainz2=cat(3,rapportgainz2,rapportgainz2(:,:,taille
(3)-1));

rapportcx=rapportc;rapportcx(taille(1),:,:)=[];rapportcx=cat(1,rapportcx(1,:,:),rapportcx);
rapportcy=rapportc;rapportcy(:,taille(2),:)=[];rapportcy=cat(2,rapportcy(:,1,:),rapportcy);
rapportcz=rapportc;rapportcz(:,:,taille(3))=[];rapportcz=cat(3,rapportcz(:,:,1),rapportcz);
rapportcx2=rapportc;rapportcx2(1,:,:)=[];rapportcx2=cat(1,rapportcx2,rapportcx2(taille(1)-1,:,:));
rapportcy2=rapportc;rapportcy2(:,1,:)=[];rapportcy2=cat(2,rapportcy2,rapportcy2(:,taille(2)-1,:));
rapportcz2=rapportc;rapportcz2(:,:,1)=[];rapportcz2=cat(3,rapportcz2,rapportcz2(:,:,taille(3)-1));
```

```
V=V-
((Qmoinsx>0).*rapportgainx.*rapport.*Qmoinsx+(Qmoinsy>0).*rapportgainy.*rapport.*Qmoinsy+(Qmoin
sz>0).*rapportgainz.*rapport.*Qmoinsz+(Qx>0).*rapportgainx2.*rapport.*Qx+(Qy>0).*rapportgainy2.*rap
port.*Qy+(Qz>0).*rapportgainz2.*rapport.*Qz+(Qmoinsx<0).*rapportgain.*rapportx.*Qmoinsx+(Qmoinsy
<0).*rapportgain.*rapporty.*Qmoinsy+(Qmoinsz<0).*rapportgain.*rapportz.*Qmoinsz+(Qx<0).*rapportgai
n.*rapportx2.*Qx+(Qy<0).*rapportgain.*rapporty2.*Qy+(Qz<0).*rapportgain.*rapportz2.*Qz)*echelletem
ps;
C=C-
(pertec.*rapportc+((Qcmoinsx<0).*Qcmoinsx.*rapportcx+(Qcmoinsy<0).*Qcmoinsy.*rapportcy+(Qcmoins
z<0).*Qcmoinsz.*rapportcz+(Qcx<0).*Qcx.*rapportcx2+(Qcy<0).*Qcy.*rapportcy2+(Qcz<0).*Qcz.*rappo
rtcz2)*echelletemps);

imageeau1=zeros(11,11,3);imageN1=imageeau1;imageeau1(:,:,3)=floor((255/30)*sum(V.*~bZ0,3));image
N1=floor((255/(((3*400*10^-3)/(3*1.5))/86800)*echelletemps*50)*sum(C.*~bZ0,3));
eval(['imwrite(imageeau1,"eaudessus1' num2str(2*i) '.bmp","BMP")'])
eval(['imwrite(imageN1,"nitratedessus1' num2str(2*i) '.bmp","BMP")'])
imageeau2=zeros(11,11,3);imageN2=imageeau2;imageeau2(:,:,3)=floor((255/30)*squeeze(sum(V.*~bZ0,2)
));imageN2(:,:,3)=floor((255/(((3*400*10^-
3)/(3*1.5))/86800)*echelletemps*50)*squeeze(sum(C.*~bZ0,2)));
eval(['imwrite(imageeau2,"eaudessus2' num2str(2*i) '.bmp","BMP")'])
eval(['imwrite(imageN2,"nitratedessus2' num2str(2*i) '.bmp","BMP")'])
end



function [points,t] = fnplt2(limite,f,varargin)
%FNPLT Plot a function.
%
%   FNPLT(F)  plots the function in F on its basic interval.
%
%   FNPLT(F,SYMBOL,INTERV,LINEWIDTH,JUMPS) plots the function F
%   on the specified INTERV = [a,b] (default is the basic interval),
%   using the specified plotting SYMBOL (default is '-'),
%   and the specified LINEWIDTH (default is 1),
%   and using NaNs in order to show any jumps as actual jumps only
%   in case JUMPS is a string beginning with 'j'.
%
%   The four optional arguments may appear in any order, with INTERV
%   the one of size [1 2], SYMBOL and JUMPS strings, and LINEWIDTH the
%   scalar. Any empty optional argument is ignored.
%
%   If the function in F is 2-vector-valued, the planar curve is
%   plotted.  If the function in F is d-vector-valued with d>2, the
%   space curve given by the first three components of F is plotted.
%
%   If the function is multivariate, it is plotted as a bivariate function
%   at the midpoint of its basic intervals in additional variables, if any.
%
%   POINTS = FNPLT(F,...)   does not plot, but returns instead the sequence
%   of 2D-points or 3D-points it would have plotted.
%
%   [POINTS,T] = FNPLT(F,...)  also returns, for a vector-valued F, the
%   corresponding vector T of parameter values.
%
%   Example:
```

```
%      x=linspace(0,2*pi,21); f = spapi(4,x,sin(x));
%      fnplt(f,'r',3,[1 3])
%
%   plots the graph of the function in f, restricted to the interval [1,3],
%   in red, with linewidth 3 .

%   Copyright 1987-2002 C. de Boor and The MathWorks, Inc.
%   $Revision: 1.20 $

% interpret the input:
symbol=''; interv=[]; linewidth=[]; jumps=0;
for j=2:2
   arg = varargin{j-1};
   if ~isempty(arg)
      if ischar(arg)
         if arg(1)=='j', jumps = 1;
         else, symbol = arg;
         end
      else
         [ignore,d] = size(arg);
         if ignore~=1, error(['arg',num2str(j),' is incorrect.']), end
         if d==1
            linewidth = arg;
         else
            interv = arg;
         end
      end
   end
end

% generate the plotting info:
if ~isstruct(f), f = fn2fm(f); end

switch f.form(1:2)
case 'st'
   if ~isempty(interv), f = stbrk(f,interv);
   else
      interv = stbrk(f,'interv');
   end
   npoints = limite; d = stbrk(f,'dim');
   switch fnbrk(f,'var')
   case 1
      x = linspace(interv{1}(1),interv{1}(2),npoints);
      v = stval(f,x);
   case 2
      x = {linspace(interv{1}(1),interv{1}(2),npoints), ...
                  linspace(interv{2}(1),interv{2}(2),npoints)};
      [xx,yy] = ndgrid(x{1},x{2});
      v = reshape(stval(f,[xx(:),yy(:)].'),[d,size(xx)]);
   otherwise
      error('Cannot handle st functions with more than 2 variables.')
   end
otherwise
   if ~strcmp(f.form([1 2]),'pp')
      givenform = f.form; f = fn2fm(f,'pp'); basicint = ppbrk(f,'interval');
```

```
end

if ~isempty(interv), f = ppbrk(f,interv); end

[breaks,coefs,l,k,d] = ppbrk(f);
if iscell(breaks)
   m = length(breaks);
   for i=m:-1:3
      x{i} = (breaks{i}(1)+breaks{i}(end))/2;
   end
   npoints = limite;
   ii = [1]; if m>1, ii = [2 1]; end
   for i=ii
      x{i}= linspace(breaks{i}(1),breaks{i}(end),npoints);
end
v = ppual(f,x);
if exist('basicint') % we converted from B-form to ppform, hence must now
                     % enforce the basic interval for the underlying spline.
   for i=ii
      temp = find(x{i}<basicint{i}(1)|x{i}>basicint{i}(2));
      if d==1
         if ~isempty(temp), v(:,temp,:) = 0; end
         v = permute(v,[2,1]);
      else
         if ~isempty(temp), v(:,:,temp,:) = 0; end
         v = permute(v,[1,3,2]);
      end
   end
   end
else     % we are dealing with a univariate spline
   npoints = 101;
   x = [breaks(2:l) linspace(breaks(1),breaks(l+1),npoints)];
   v = ppual(f,x);
   if l>1 % make sure of proper treatment at jumps if so required
      if jumps
         tx = breaks(2:l); temp = repmat(NaN, d,l-1);
      else
         tx = []; temp = zeros(d,0);
      end
      x = [breaks(2:l) tx x];
      v = [ppual(f,breaks(2:l),'left') temp v];
   end
   [x,inx] = sort(x); v = v(:,inx);

if exist('basicint') % we converted from B-form to ppform, hence must now
                     % enforce the basic interval for the underlying spline.
                     % Note that only the first d components are set to zero
                     % outside the basic interval, i.e., the (d+1)st
                     % component of a rational spline is left unaltered :-)
   if jumps, extrap = repmat(NaN,d,1); else, extrap = zeros(d,1); end
   temp = find(x<basicint(1)); ltp = length(temp);
   if ltp
      x = [x(temp),basicint([1 1]), x(ltp+1:end)];
      v = [zeros(d,ltp+1),extrap,v(:,ltp+1:end)];
   end
```

```
      temp = find(x>basicint(2)); ltp = length(temp);
      if ltp
        x = [x(1:temp(1)-1),basicint([2 2]),x(temp)];
        v = [v(:,1:temp(1)-1),extrap,zeros(d,ltp+1)];
      end
      %   temp = find(x<basicint(1)|x>basicint(2));
      %   if ~isempty(temp), v(temp) = zeros(d,length(temp)); end
    end
end

if exist('givenform')&givenform(1)=='r' % we are dealing with a rational fn:
                                        % need to divide by last component
    d = d-1;
      sizev = size(v); sizev(1) = d;
      % since fnval will replace any zero value of the denominator by 1,
      % so must we here, for consistency:
      v(d+1,find(v(d+1,:)==0)) = 1;
      v = reshape(v(1:d,:)./repmat(v(d+1,:),d,1),sizev);
    end
end

%  use the plotting info, to plot or else to output:
if nargout==0
    if isempty(symbol), symbol = '-'; end
    if isempty(linewidth), linewidth=1; end
    if iscell(x)
      switch d
        case 1
      [yy,xx] = meshgrid(x{2},x{1});
      surf(xx,yy,reshape(v,length(x{1}),length(x{2})))
    case 2
      v = squeeze(v); roughp = 1+(npoints-1)/5;
      vv = reshape(cat(1,...
          permute(v(:,1:5:npoints,:),[3,2,1]),...
          repmat(NaN,[1,roughp,2]),...
          permute(v(:,:,1:5:npoints),[2,3,1]),...
          repmat(NaN,[1,roughp,2])), ...
              [2*roughp*(npoints+1),2]);
      plot(vv(:,1),vv(:,2))
    case 3
      v = permute(reshape(v,[3,length(x{1}),length(x{2})]),[2 3 1]);
      surf(v(:,:,1),v(:,:,2),v(:,:,3))
  otherwise
      end
    else
      switch d
        case 1, plot(x,v,symbol,'linew',linewidth)
        case 2, plot(v(1,:),v(2,:),symbol,'linew',linewidth)
        otherwise, plot3(v(1,:),v(2,:),v(3,:),symbol,'linew',linewidth)
      end
    end
  else
    if iscell(x)
      switch d
      case 1
```

```
    [yy,xx] = meshgrid(x{2},x{1});
    points = {xx,yy,reshape(v,length(x{1}),length(x{2}))};
  case 2
    [yy,xx] = meshgrid(x{2},x{1});
    points = {xx,yy,reshape(v,[2,length(x{1}),length(x{2})])};
  case 3
    points = {squeeze(v(1,:)),squeeze(v(2,:)),squeeze(v(3,:))};
    t = {x{1:2}}
  otherwise
  end
else
  if d==1, points = [x;v];
  else, t = x; points = v([1:min([d,3])],:); end
  end
end
```

%script_cartegeol2 ; lecture de la carte géologique.

```
texte=input('entrez le chemin de la légende','s');
eval(['P=imread(' texte ');'])
taille=size(P);
coordonneeX=[];
coordonneeY=[];

Y=sum(P(:,floor(taille(2)/2),:),3);
Y=(Y~=0);Y1=[Y(length(Y));Y(1:length(Y)-1)];Y2=[Y(2:length(Y));Y(1)];
A=(Y1&Y2);
Y=Y-A;
coordonneeX=rot90(sort([find(Y);find(Y)]));

for i=1:2:length(coordonneeX) %boucle permettant d'établir l'abscisse de chaque limite de figuré de la
légende.
   Y=sum(P(coordonneeX(i),:,:),3);
   Y=(Y~=0);Y1=[Y(length(Y)) Y(1:length(Y)-1)];Y2=[Y(2:length(Y)) Y(i)];
   A=(Y1&Y2);
   Y=Y-A;
   coordonneeY=[coordonneeY find(Y)];
end
coordonnee=[coordonneeX;coordonneeY];compteur=0;
for i=1:4:length(coordonneeX) %boucle permettant de définir l'intervalle de couleur représentatif de chaque
affleurement.
   compteur=compteur+1;
   eval(['RGB' num2str(compteur) '=[];Couche' num2str(compteur) '=[];Couche' num2str(compteur)
'lim=[];'])
   A=P(coordonnee(1,i):coordonnee(1,i+2),coordonnee(2,i):coordonnee(2,i+1),:);  %on limite l'examen au
domaine de la légende correspondant au figuré de numéro i.
   eval(['RGB' num2str(compteur) '=double([min(min(A(:,:,1))) min(min(A(:,:,2)))
min(min(A(:,:,3)));max(max(A(:,:,1))) max(max(A(:,:,2))) max(max(A(:,:,3)))]);'])
   eval(['Couche' num2str(compteur+1) '=[];Couche' num2str(compteur+1) 'lim=[];'])
end
texte=input('entrez le chemin de la carte géologique','s');
eval(['I=imread(' texte ');'])
taille=size(I);
for k=1:length(coordonneeY)/4%on va de la couche la plus jeune à la plus ancienne.
```

```
    RGBmin=double(zeros(size(I)));
    RGBmax=double(zeros(size(I)));
    eval(['RGBmin(:,:,1)=RGBmin(:,:,1)+RGB' num2str(k) '(1,1);RGBmax(:,:,1)=RGBmax(:,:,1)+RGB'
num2str(k) '(2,1);'])
    eval(['RGBmin(:,:,2)=RGBmin(:,:,2)+RGB' num2str(k) '(1,2);RGBmax(:,:,2)=RGBmax(:,:,2)+RGB'
num2str(k) '(2,2);'])
    eval(['RGBmin(:,:,3)=RGBmin(:,:,3)+RGB' num2str(k) '(1,3);RGBmax(:,:,3)=RGBmax(:,:,3)+RGB'
num2str(k) '(2,3);'])
    A=(sum(((I>=RGBmin)&(I<=RGBmax)),3)==3);
    %[X Y]=find(A);
    %eval(['Couche' num2str(k) '=[Couche' num2str(k) ' [X;Y]];'])
    %eval(['Y=Couche' num2str(k) ';RGB=RGB' num2str(k) ';'])
    B=(sum((I<=(zeros(size(I))+10)),3)==3);
    Ah1=A;Ah1=[Ah1 zeros(size(Ah1(:,1:3,:)))];Ah1(:,1:3,:)=[];
    Ah2=A;Ah2=[zeros(size(Ah2(:,taille(2)-2:taille(2),:))) Ah2];Ah2(:,taille(2):taille(2)+2,:)=[];
    Av1=A;Av1=[Av1;zeros(size(Av1(1:3,:,:)))];Av1(1:3,:,:)=[];
    Av2=A;Av2=[zeros(size(Av2(taille(1)-2:taille(1),:,:)));Av2];Av2(taille(1):taille(1)+2,:,:)=[];
    C=B&((Ah1+Ah2+Av1+Av2)~=0);
    [X,Y]=find(C);
    R=[rot90(X);rot90(Y)];
    %R(2,:)=745-R(2,:);
    %R=filtre(R);
    eval(['Couche' num2str(k) 'lim=R;'])
    for i=1:length(X)
       I(X(i),Y(i),:)=[255 255 255];
    end
            % ainsi une limite de couche n'est prise en compte qu'une fois.
end

%on applique ensuite une méthode pour réaliser des coupes géologiques en
%supposant que les limites de couche sont des plans.
%for i=1:length(coordonneeY)/4+1
%    eval(['DV=(Couche' num2str(i) 'lim);'])
%    [a,v]=sort(DV(2,:),2);
%    DH=DV(:,v);
%    [a,v]=sort(DV(1,:),2);
%    DV=DV(:,v);
%    ZV=Z;ZV(:,:)=0;ZH=Z;ZH(:,:)=0;
%    for j=1:length(DV)
%        A=DV(:,j);DV(:,j)=[];C=DV(:,j+1);a=C(1,1)-A(1,1);
%        B=DH(:,j);DH(:,j)=[];D=DH(:,j+1);b=D(2,1)-B(2,1);
%        X=zeros(a);T=X(:,1);T(1,1)=Z(A(1,1),A(2,1));T(a,1)=Z(C(1,1),C(2,1));
%        ZV(A(1,1):C(1,1),A(2,1))=egalise(T);
%        Y=zeros(b);S=Y(1,:);S(1,1)=Z(D(1,1),B(2,1));S(b,1)=Z(B(1,1),D(2,1));
%        ZH(B(1,1),B(2,1):D(2,1))=egalise(S);
%    end
%    eval(['Z' num2str(i) '=(ZV+ZH)/2;'])
%end
%autre méthode utilisant les B-splines : considérer les coordonnées des
%points des limites d'affleurement comme des noeuds et la valeur de
%l'altitude en ces points comme la valeur des noeuds :

% hold on
 for i=1:length(coordonneeY)/2
     Q=eval(['Couche' num2str(i) 'lim']);altitude=[];
```

```
    for k=1:length(Q)
        W=Q(:,k);altitude=[altitude Z(W(1),W(2))];
    end
    j = tpaps(Q,altitude,1);fig1=fnplt2(744,j,2);
    eval(['Z' num2str(i) '=cell2mat(fig1(3));'])
 end

Z0=Z;echelle=input('entrez la valeur en mètres d''un pixel ');
maximum=max(max(Z0))+10;
minimum=0;

%définition des caractéristiques de l'air
z1=Z0;taille=size(z1);
x=linspace(1,floor((taille(1)-1)/4)*4+1,floor(taille(1)/4));y=linspace(1,floor((taille(2)-
1)/4)*4+1,floor(taille(2)/4));z=linspace(1,floor(ceil(maximum)/4)*4+1,ceil(maximum/4)+1);
   z1=z1(x,:,:);z1=z1(:,y,:);
z2=zeros(size(z1))+ceil(maximum);
z3=zeros(taille(1)/4,taille(2)/4,ceil(maximum/4)+1);compteur=0;a=z3;b=a;
for i=ceil(maximum):-4:0
   compteur=compteur+1;a(:,:,compteur)=z1;b(:,:,compteur)=z2;
   z3(:,:,compteur)=i;
end
bZ0=double((z3<=b)&(z3>=a));
K=bZ0;rho=zeros(size(K));h=rho;couche=h;

porosite=0;nb_couche=7;g=9.8;
%définitions des caractéristiques des couches
for k=3:nb_couche+3
   z1=eval(['Z' num2str(k)]);taille=size(Z0);
   x=linspace(1,floor((taille(1)-1)/4)*4+1,floor(taille(1)/4));y=linspace(1,floor((taille(2)-
1)/4)*4+1,floor(taille(2)/4));z=linspace(1,floor(ceil(maximum)/4)*4+1,ceil(maximum/4)+1);
   z1=z1(x,:,:);z1=z1(:,y,:);
   z2=eval(['Z' num2str(k-1)]);taille=size(Z0);
   z2=z2(x,:,:);z2=z2(:,y,:);
   z3=zeros(taille(1)/4,taille(2)/4,ceil(maximum/4)+1);compteur=0;a=z3;b=a;
   for i=ceil(maximum):-4:0
      compteur=compteur+1;a(:,:,compteur)=z1;b(:,:,compteur)=z2;
      z3(:,:,compteur)=i;
   end
   eval(['bZ' num2str(k) '=double((z3<=b)&(z3>=a));'])
   permeabilite=input(['entrez la permeabilité de la couche numéro ' num2str(k)]);
   K=K+permeabilite*eval(['bZ' num2str(k)]);
   masse_volumique=input(['entrez la masse volumique de la couche numéro ' num2str(k)]);
   rho=rho+masse_volumique.*eval(['bZ' num2str(k)]);
   h=b-z3.*eval(['bZ' num2str(k)])+h;couche=couche+k*eval(['bZ' num2str(k)]);
   eval(['couche_info' num2str(k) '={z1,z2,bZ' num2str(k) ',permeabilite,masse_volumique,porosite};'])
end
P=g*rho.*h;taille=size(P);
Pxplus4=P;Pxplus4(1,:,:)=[];Pxplus4=cat(1,Pxplus4,P(taille(1),:,:));
Pyplus4=P;Pyplus4(:,1,:)=[];Pyplus4=cat(2,Pyplus4,P(:,taille(2),:));
Pzplus4=P;Pzplus4(:,:,taille(3))=[];Pzplus4=cat(3,Pzplus4,P(:,:,1));
dPsurdx=Pxplus4-P;dPsurdy=Pyplus4-P;dPsurdz=1+Pzplus4-P;
Kx1=K;Kx1(1,:,:)=[];Kx1=cat(1,Kx1,K(taille(1),:,:));hx1=h;hx1(1,:,:)=[];hx1=cat(1,hx1,h(taille(1),:,:));Kx=
(K.*h+Kx1.*hx1)./(h+hx1);
```

```
Ky1=K;Ky1(:,1,:)=[];Ky1=cat(2,Ky1,K(:,taille(2),:));hy1=h;hy1(:,1,:)=[];hy1=cat(2,hy1,h(:,taille(2),:));Ky=
(K.*h+Ky1.*hy1)./(h+hy1);
Kz1=K;Kz1(:,:,taille(3))=[];Kz1=cat(3,Kz1,K(:,:,1));hz1=h;hz1(:,:,taille(3))=[];hz1=cat(3,hz1,h(:,:,1));Kz=(
h+hz1)./((h./K)+(hz1./Kz1));
qx=-Kx.*dPsurdx;qy=-Ky.*dPsurdy;qz=-Kz.*dPsurdz;
Qx=(echelle^2)*qx;Qy=(echelle^2)*qy;Qz=(echelle^2)*qz;
```

%modélisation d'un ensemble d'une couche n'apparaissant que dans la zone
%étudiée. (ici la couche 6)

```
for k=3:9
    H2=eval(['Couche' num2str(k) 'limp']);
    xmin=min(H2(1,:))
    xmax=max(H2(1,:))
    ymin=min(H2(2,:))
    ymax=max(H2(2,:))
    H=[H2 eval(['Couche' num2str(k)])];

    H(:,find(H(1,:)<xmin))=[];
    H(:,find(H(1,:)>xmax))=[];
    H(:,find(H(2,:)<ymin))=[];
    H(:,find(H(2,:)>ymax))=[];

    H(2,:)=H(2,:)-ymin+1;
    H(1,:)=H(1,:)-xmin+1;
    if max(H(1,:))>max(H(2,:))
        H(2,:)=H(2,:)+max(H(1,:))-max(H(2,:));
    else
        H(1,:)=H(1,:)+max(H(2,:))-max(H(1,:));
    end
    j=tpaps(H(1:2,:),H(3,:));
    fig1=fnplt2(max(max(H(1,:)),max(H(2,:))),j,2);
    W=cell2mat(fig1(3));
    eval(['Z' num2str(k) '=Z;']);
    for i=1:max(max(H(1,:)),max(H(2,:)))
        if i+xmin-1>744
            break
        else
        eval(['Z' num2str(k) '(i+xmin-1,ymin:ymin+max(max(H(1,:)),max(H(2,:)))-1)=W(i,:);']);
    end
    end
    T=Z<eval(['Z' num2str(k)]);
    eval(['Z' num2str(k) '=Z' num2str(k) '.*~T+Z.*T;']);
    end
    for i=744:-1:650
    a=find(Couche6limp(1,:)==i);size(a)
    Z6(1:min(a(1,:)),:)=Z(1:min(a(1,:)),:);
    end
```

%récupération des altitudes des points de l'affleurement des limites de
%couche.

```
for k=1:8
```

```
    H=eval(['Couche' num2str(k) 'lim']);
    altitude=[];
    for i=1:length(H)
        altitude=[altitude Z(H(1,i),H(2,i))];
    end
    eval(['Couche' num2str(k) 'lim=[Couche' num2str(k) 'lim;altitude];'])
end
```

%calcul des pendages le long d'un trait de coupe :

```
function pendage=calcul-pendage_colonne(k)
for i=3:9
H=eval(['Couche' num2str(i) 'limp']);
eval(['a' num2str(i) '=H(:,find(H(2,:)==k));'])
end
for i=3:9
H=eval(['Couche' num2str(i) 'limp']);
eval(['a2' num2str(i) '=H(:,find(H(2,:)==k+1));'])
end
pendage=[];
for i=3:8
taille=eval(['[size(a' num2str(i) ') size(a' num2str(i+1) ')]']);
if min(taille)~=0
a=eval(['[a' num2str(i) '(:,1) a2' num2str(i) '(:,1)]']);
b=eval(['[a' num2str(i+1) '(:,1) a2' num2str(i+1) '(:,1)]']);
dist1=sqrt((a(1,1)-b(1,1))^2+(a(2,1)-b(2,1))^2+(a(3,1)-b(3,1))^2);
dist2=sqrt((a(1,2)-b(1,2))^2+(a(2,2)-b(2,2))^2+(a(3,2)-b(3,2))^2);
y1=a(3,1)-b(3,1);y2=a(3,2)-b(3,2);
pendage=[pendage abs(atan((y1-y2)/((dist2-dist1)*echelle)))];
else
pendage=[pendage 00];
end
end
```

%approximation polynomiale de degré 3 d'une couche dans un plan de coupe.

```
f=Z(:,k);
x=0:max(a9(1,:))-min(a9(1,:));
fonction=f(min(a9(1,:)):max(a9(1,:)));
couche1=-tan(pendage(j))*x+fonction(1);
couche2=tan(pendage(j+1))*x-tan(pendage(j))*x(length(x))+fonction(length(fonction));
coeffdir2=-tan(pendage(j));
coeffdir1=tan(pendage(j+1))

delta=fonction(1);
gamma=coeffdir2;
beta=(3*(fonction(length(fonction))-delta)-x(length(x))*(2*gamma+coeffdir1))/(x(length(x))^2);
alpha=(fonction(length(fonction))-x(length(x))*gamma-delta)/(x(length(x))^3)-beta/(x(length(x)));

fonction2=alpha*x.^3+beta*x.^2+gamma*x+delta;

couche=f;
couche(min(a9(1,:)):max(a9(1,:)))=fonction2;
```

%calcul de l'épaisseur des couches et de leurs pendages à différents points

```
H2=eval(['Couche' num2str(k) 'lim']);
H=eval(['Couche' num2str(k-1) 'lim']);
for i=1:length(H)
    a=H(:,i);
    c=H2(:,find(H2(2,:)==a(2)));
    if length(c)>=1
    b=H2(:,find(sqrt((c(1)-a(1)).^2+(c(2)-a(2)).^2+(c(3)-a(3)).^2)==min(sqrt((c(1)-a(1)).^2+(c(2)-
a(2)).^2+(c(3)-a(3)).^2))));
    epaisseur_apparente=sqrt((b(1)-a(1)).^2+(b(2)-a(2)).^2+(b(3)-a(3)).^2);
    y=abs(a(3)-b(3));

    r=find((H(2,:)==a(2)+1)&(abs(a(1)-H(1,:))<=2));
    if length(r)>=1
    a2=H(:,r(1));
    c2=H2(:,find(H2(2,:)==a2(2)));
    if length(c2)>=1
       b2=H2(:,find(sqrt((c2(1)-a2(1)).^2+(c2(2)-a2(2)).^2+(c2(3)-a2(3)).^2)==min(sqrt((c2(1)-
a2(1)).^2+(c2(2)-a2(2)).^2+(c2(3)-a2(3)).^2))));
       epaisseur_apparente2=sqrt((b2(1)-a2(1)).^2+(b2(2)-a2(2)).^2+(b2(3)-a2(3)).^2);
       y2=abs(a2(3)-b2(3));
       donnees=[donnees;[epaisseur_apparente,y,i,f];[epaisseur_apparente2,y2,i+1,f2]];
       coord=[coord [a(1);a(2);a(3)]];e=[e epaisseur_apparente];diff=[diff y];
    end
    end
end
end

taille2=size(donnees);epaisseurs_apparentes=[];ys=[];
for i=1:2:taille2(1)-1
    epaisseurs_apparentes=[epaisseurs_apparentes (donnees(i,1)-donnees(i+1,1))];
    ys=[ys (donnees(i+1,2)-donnees(i,2))];
end
pendage=abs(atan(ys./epaisseurs_apparentes));
epaisseur=abs(e.*sin(pendage)+diff.*cos(pendage));
g=find(epaisseur<(mean(epaisseur)/4));
epaisseur(g)=[];pendage(g)=[];coord(:,g)=[];
nv_coord=coord;
e_vert=(epaisseur./abs(cos(pendage)));
g=find(e_vert>(mean(e_vert)+mean(e_vert)*(25/100)));
nv_coord(3,:)=nv_coord(3,:)-abs(e_vert);
nv_coord(:,g)=[];
eval(['Couche' num2str(k) 'lim=[Couche' num2str(k) 'lim nv_coord];'])
eval(['Couche' num2str(k) 'data=[pendage;epaisseur];'])
```

%modélisation d'une couche en 3D en tenant compte des données :

```
for i=1:165
matrice=eval(G(i,:));
t=H(3,matrice(1));
H(3,matrice)=max(t);
end
saut=input('entrez le saut')
```

```
for i=1:165
    matrice=eval(G(i,:));
    altitude=zeros(1,length(matrice));
    for j=1:saut:length(matrice)
        altitude(j)=H(3,matrice(j));
    end
    H(3,matrice)=altitude;
end
r=find(H(3,:)==0);
H(:,r)=[];
r=find(H(3,:)>195);
H(:,r)=[];
K='';
for i=1:165
    matrice=eval(G(i,:));
    r=find(matrice>length(H));
    matrice(r)=[];
    K=strvcat(K,mat2str(matrice));
end
G=K;
T=[-3 -7 -8 -2 99999999];nombre=size(G);
while (T(length(T))~=T(length(T)-4))&(T(length(T))>5000)
aeliminer=[];
for i=1:nombre(1)
    matrice=eval(G(i,:));
    F2=matrice;
    longueur=length(F2);
    compteur=0;
    K=[-3 -7 -8 -2 -1];
    while (K(length(K))~=K(length(K)-4))&(compteur<longueur)
        t=H(:,F2(1));
        F2(1)=[];
        f2=find(sqrt(abs(H(1,F2)-t(1)).^2+abs(H(2,F2)-t(2)).^2)<2.5);
        aeliminer=[aeliminer F2(f2)];
        F2(f2)=[];
        longueur=length(F2);
        K=[K longueur];
    end
end
H(:,aeliminer)=[];
T=[T length(H)];
G=regroupement2(H);nombre=size(G);
end
```

%visualisation d'une matrice de même format que H.

```
function visualiseH(H,u)
E=zeros(u);
for i=1:length(H)
    E(H(1,i),H(2,i))=100;
end
mesh(E)
```

%calcul et visualisation de l'évolution des volumes d'eau ruisselantes en fonction du temps
%unité de temps : 1 jour
%chaque unité de surface a une inclinaison ici négligée, seule la pente entre deux unités de surface est prise en compte

%Données_____
_____
% Z : matrice altitude
% R : martice coefficient de ruissellement
% taille : size(Z)
% apport : quantité de pluie en 1 mois (vecteur (1x12))
% jours : nombre de jours par mois (vecteur (1x12))
% echelle : valeur en mètres de la maille
%_____
_____
E=imread('carte.bmp');
R=double(E(:,:,1))/100;
%on entre la taille en metres de la maille
echelle=8.3;
compteur=0;
%initialisation de la variable jours pour une année non bissextile
jours=[31,28,31,30,31,30,31,31,30,31,30,31];

%Entrée des coefficients de ruissellement, parcelle par parcelle, selon le type de terrain présent
%a determiner
taille=size(Z)
% Calcul des pentes en tout point
pente_h=Z;
pente_h(1,:)=[];
pente_h=[pente_h;pente_h(taille(1)-1,:)];
%On décale les lignes d'une unité vers le haut pour déterminer la variation d'eau (apport ou perte)
pente_h=Z-pente_h;   %On calcule la pente sur les horizontales

pente_v=Z;
pente_v(:,1)=[];
pente_v=[pente_v,pente_v(:,taille(2)-1)];
%On décale les colonnes d'une unité vers la gauche
pente_v=Z-pente_v; %on calcule la pente sur les verticales

%On décale de meme vers la droite et vers la bas

pente_h2=Z;
pente_h2(taille(1),:)=[];
pente_h2=[pente_h2(1,:);pente_h2];
%On décale les lignes d'une unité vers le bas pour déterminer la variation d'eau (apport ou perte)
pente_h2=Z-pente_h2;   %On calcule la pente sur les horizontales

pente_v2=Z;
pente_v2(:,taille(2))=[];
pente_v2=[pente_v2(:,1),pente_v2];
%On décale les colonnes d'une unité vers la droite
pente_v2=Z-pente_v2;   %on calcule la pente sur les verticales

% On ne tient pas compte de la pente diagonale

```
date=input('Entrez la date du premier jour sous la forme [jour,mois] : ')
v=0;
stop=0;apport=input('Entrez la pluviometrie mois par mois (sur un an) sous forme de vecteur : ' );
while stop==0    %Mise à jour de la date
    compteur=compteur+1;
    if date(1)==jours(date(2))     %si on est au dernier jour du mois
        date(1)=1;                 %on passe au premier jour du mois suivant
        date(2)=(date(2)*abs(floor(date(2)/12)-1))+1;  % abs(floor(date(2)/12)-1) vaut 0 si on est au douzieme
mois et 1 sinon
    else
        date(1)=date(1)+1;         %si on n'est pas à la fin du mois, on rajoute un au jour
    end

    apport=apport.*jours;
    pluie=apport(date(2))/jours(date(2));   % on calcule le volume d'eau tombée en moyenne sur le mois
    v=v+pluie*echelle^2;    %on ajoute à v (initialisé à 0) le volume de pluie
    v=v-(1-R).*v;   %on soustrait à v l'eau s'infiltrant

pente_totale=(pente_h>0).*pente_h+(pente_h2>0).*pente_h2+(pente_v>0).*pente_v+(pente_v2>0).*pente_
v2+10^-99;
    % on ajoute 10^-99 (ce qui est négligeable) car il nous aurait fallu diviser par 0 ce qui est impossible

    A=(pente_totale~=0);

    %on calcule le volume d'eau transférée vers le bas
    transfert_h=v.*(pente_h./pente_totale);

%on calcule le volume d'eau transférée vers la droite
transfert_v=v.*(pente_v./pente_totale);

%on calcule le volume d'eau transférée vers le haut
transfert_h2=v.*(pente_h2./pente_totale);

%on calcule le volume d'eau transférée vers la gauche
transfert_v2=v.*(pente_v2./pente_totale);

%on soustrait à v l'eau ruisselante
v=v-(transfert_h+transfert_h2+transfert_v+transfert_v2).*A;
%on multiplie par A pour supprimer les pentes négatives (puisque l'eau ne peut pas remonter une pente)

%on décale vers le bas
transfert_h(taille(1),:)=[];
transfert_h=[transfert_h(1,:);transfert_h]; %concaténation verticale

%On décale vers le haut
transfert_h2(1,:)=[];
transfert_h2=[transfert_h2;transfert_h2(taille(1)-1,:)];  %concaténation verticale

%on décale vers la droite
transfert_v(:,taille(2))=[];
transfert_v=[transfert_v(:,1),transfert_v];     %concaténation horizontale

%on décale vers la gauche
transfert_v2(:,1)=[];
```

```
transfert_v2=[transfert_v2,transfert_v2(:,taille(2)-1)];  %concaténation horizontale

%ajout à v de l'apport d'eau par les cases aux alentour
v=v+(transfert_h+transfert_h2+transfert_v+transfert_v2).*(~A);

hauteur_d_eau=v/(echelle^2);                %on prefere visualiser la hauteur d'eau sur chaque parcelle,
eau=E;
eau(:,:,:)=255;
eau(:,:,1)=hauteur_d_eau*70;                                % plutot que le volume en metres cube
image(eau)
if floor(compteur/2)~=compteur/2
    imwrite(eau,['inondation' num2str(date(1)) num2str(date(2)) '.bmp'],'BMP')
end
    stop=0;
    if compteur>=30
        break
    end
end
%hold off
%axis equal
%mesh(Z+hauteur_d_eau,C)
%C est la couleur à choisir pour le graphe (comme on représente de l'eau, on choisit 'b' (blue=bleu)
```

%écoulement et transport sans apport d'azote.

```
% modélisation de l'écoulement.
%donnees : Qx,Qy,Qz,Z
taille=size(Qx);
pluvio=([0.6 0 0.4 7.4 0.2 2.6 0 3.2 0.2 0.2 0.2 0 1.6 0.6 0.2 0 0 0 0 0.2 1.6 4.6 0.8 0 0.2 0 0 6.6 1.8 0 0.2 0.2
3.4 0 0 0 1.6 5.2 11.2 0.2 7.4 0.2 6.2 5.4 3.8 5.2 3.0 3.2 3.4 0.2 0.2 0.4 0 0.8 2.0 1.6 0.2 0 5.8 3.2 5.6 0 14.6 0
0 0.2 1.2 1.8 0 0 0 0 0 3.8 0 0 0 0.6 0 0 0.2 0 0 2.4 2.2 2.0 3.0 0 .0 14.2 9.2 0.4 0 1.6 5.6 1.8 0 0 0.2 0 0 0.6
0.6 3.2 4.8 1 2 8.2 0 0.2 1.8 6.4 12 3.8 0.2 0 0.4 7.6 13.2 0.6 0 0 0 0 0.2 0 2.8 0.6 0 0 12.6 59.2 0.4 0.2 0.2 0
0.6 1.4 1.8 3.8 0.2 5.2 0 1 1 5.2 1 5.4 0.2 6 11 0 0.2 0.2 21.8 3 14 2.2 0.2 0 0 0 0 0.2 0 0 0 0 0 0 0 0 1.6 0 0.4
0.4 0 0 0 0 0 1 0 44 0 9.2 0.8 0.2 13.4 5.8 8.2 14.8 0.6 0 6.6 7.2 6. 2.6 0.2 0 0 0 0 0.2 21.2 7.8 2.4 10.6 0.2
1.2 0 0 5 1.6 15.4 5.4 0.2 0 0 2 0.2 0 0.2 0 1.4 0 0 0.2 0 0.2 2.2 0 0.2 0 0 0 0 3.6 0.2 0.2 0 0 0.4 9.2 14 0.6 0
1.4 10 0 0.6 0 0.2 0.2 0 0 0.2 5. 1.8 0.2 0 4.2 0.2 0 0.2 0 0.4 0 0 1 5.8 4.6 0.2 0.2 3.6 0 0.2 0.2 0.2 3 11.8 15.6
10.6 0.2 6.2 0.2 0.2 7.6 12.6 0.2 15.4 0.2 0.2 0 0 0.4 0 1.6 0.4 0.2 0.2 0.2 6.4 5.8 2 2.2 0.4 0.2 9 3.8 12.2 1 0
0 0 11.6 0.6 2.2 0 8.2 0.2 1.2 5.2 0 .6 1.8 3.2 0.8 22 0 4. 0 0.6 0 5 4.2 0 1.6 1 3.8 3.2 0.2 3.8 3.6 2.4 3.8 5. 9.4
0.8 1.4 1 2.8 0 0 0 0 0 0 9.6 3 2.4 0.6 0.2 0.2 9.6]/1000)*echelle^2;
echelletemps=input('entrez l''échelle de temps ');
date=input('entrez la date sous la forme [jour mois] ');
modeleK=input('entrez 0 pour le modèle de perméabilité à non saturation de Van Genuchten, 1 pour celui de
Campbell et Brooks. ');
V=bZ0*pluvio(date(1)+1)+~bZ0.*(n-ne);%load Darcy.mat
C=V;C(:,:,:)=0;load x.mat
stop=0;
Qsx=Qx;Qsy=Qy;Qsz=Qz;
for i=1:70
    date(1)=date(1)+4;
    heau=V/(echelle^2);
P=g*1000*~bZ0.*heau;
Pxplus4=P;Pxplus4(1,:,:)=[];Pxplus4=cat(1,Pxplus4,P(taille(1),:,:));
Pyplus4=P;Pyplus4(:,1,:)=[];Pyplus4=cat(2,Pyplus4,P(:,taille(2),:));
```

```
Pzplus4=P;Pzplus4(:,:,taille(3))=[];Pzplus4=cat(3,P(:,:,1),Pzplus4);
dPsurdx=Pxplus4-P;dPsurdy=Pyplus4-P;dPsurdz=Pzplus4-P;
if modeleK==0
    KthetasurKs=sqrt(((V./(echelle^3))-(ne*(echelle^3))./((n*(echelle^3))-(ne*(echelle^3)))).*(1-(1-
(((V./(echelle^3))-(ne*(echelle^3))./((n*(echelle^3))-(ne*(echelle^3)))).^(n2./(n2-1))).^((n2-1)./n2)).^2;
else
    KthetasurKs=((V/(echelle^3))./((V~=0).*(n*(echelle^3))+(V==0).*1)).^((V~=0).*((1/(n2-
1))+3)+(V==0));
end
disp(max(max(max(V))))
disp(max(max(max(C))))
Qx=KthetasurKs.*Qsx-KthetasurKs.*K.*dPsurdx*echelle^2;
Qx=KthetasurKs.*Qsy-KthetasurKs.*K.*dPsurdy*echelle^2;
Qx=KthetasurKs.*Qsz-KthetasurKs.*K.*dPsurdz*echelle^2;
    disp(['itération ' num2str(i) ', pluviométrie à ce jour : ' num2str(pluvio(date(1))) ' '])
    V=V.*~bZ0+(sum(pluvio(date(1)-3:date(1))))*bZ0;
    Qcx=Qx.*C;Qcy=Qy.*C;Qcz=Qz.*C;
taille=size(Qx);
Qmoinsx=-Qx;
Qmoinsx(taille(1),:,:)=[];
Qmoinsx=cat(1,Qmoinsx(1,:,:),Qmoinsx);

Qcmoinsx=-Qcx;
Qcmoinsx(taille(1),:,:)=[];
Qcmoinsx=cat(1,Qcmoinsx(1,:,:),Qcmoinsx);
Qmoinsy=-Qy;
Qmoinsy(:,taille(2),:)=[];
Qmoinsy=cat(2,Qmoinsy(:,1,:),Qmoinsy);
Qcmoinsy=-Qcy;
Qcmoinsy(:,taille(2),:)=[];
Qcmoinsy=cat(2,Qcmoinsy(:,1,:),Qcmoinsy);
Qmoinsz=-Qz;
Qmoinsz(:,:,taille(3))=[];
Qmoinsz=cat(3,Qmoinsz(:,:,1),Qmoinsz);
Qcmoinsz=-Qcz;
Qcmoinsz(:,:,taille(3))=[];
Qcmoinsz=cat(3,Qcmoinsz(:,:,1),Qcmoinsz);

Qcmoinsx2=Qmoinsx.*C;Qcmoinsy2=Qmoinsy.*C;Qcmoinsz2=Qmoinsz.*C;


perte=((Qmoinsx>0).*Qmoinsx+(Qmoinsy>0).*Qmoinsy+(Qmoinsz>0).*Qmoinsz+(Qx>0).*Qx+(Qy>0).*
Qy+(Qz>0).*Qz)*echelletemps;
    rapport=abs((((max((V-(n-ne)*(echelle^3)),0)-perte)<0).*abs(max((V-(n-
ne)*(echelle^3)),0)./(perte.*(perte~=0)+(max((V-(n-ne)*(echelle^3)),0)~=0).*max((V-(n-
ne)*(echelle^3)),0).*(perte==0)+(max((V-(n-ne)*(echelle^3)),0)==0).*(perte==0)))+((max((V-(n-
ne)*(echelle^3)),0)-perte)>=0)));

pertec=((Qcmoinsx2>0).*Qcmoinsx2+(Qcmoinsy2>0).*Qcmoinsy2+(Qcmoinsz2>0).*Qcmoinsz2+(Qcx>0)
.*Qcx+(Qcy>0).*Qcy+(Qcz>0).*Qcz)*echelletemps;
rapportc=((C~=0).*abs(((C-
pertec)<0).*abs(C./(pertec.*(pertec~=0)+(C~=0).*C.*(pertec==0)+(C==0).*(pertec==0)))+((C-
pertec)>=0)));

rapportx=rapport;rapportx(taille(1),:,:)=[];rapportx=cat(1,rapportx(1,:,:),rapportx);
```

```
rapporty=rapport;rapporty(:,taille(2),:)=[];rapporty=cat(2,rapporty(:,1,:),rapporty);
rapportz=rapport;rapportz(:,:,taille(3))=[];rapportz=cat(3,rapportz(:,:,1),rapportz);
rapportx2=rapport;rapportx2(1,:,:)=[];rapportx2=cat(1,rapportx2,rapportx2(taille(1)-1,:,:));
rapporty2=rapport;rapporty2(:,1,:)=[];rapporty2=cat(2,rapporty2,rapporty2(:,taille(2)-1,:));
rapportz2=rapport;rapportz2(:,:,1)=[];rapportz2=cat(3,rapportz2,rapportz2(:,:,taille(3)-1));

gain=((Qmoinsx<0).*rapportx.*Qmoinsx+(Qmoinsy<0).*rapporty.*Qmoinsy+(Qmoinsz<0).*rapportz.*Qm
oinsz+(Qx<0).*rapportx2.*Qx+(Qy<0).*rapporty2.*Qy+(Qz<0).*rapportz2.*Qz)*echelletemps;
rapportgain=abs(((V-gain)>(n*echelle^3)).*((gain~=0).*(((n*echelle^3)-
V)./((gain)+(gain==0))+(gain==0)))+((V-gain)<=(n*echelle^3)));

rapportgainx=rapportgain;rapportgainx(taille(1),:,:)=[];rapportgainx=cat(1,rapportgainx(1,:,:),rapportgainx);
rapportgainy=rapportgain;rapportgainy(:,taille(2),:)=[];rapportgainy=cat(2,rapportgainy(:,1,:),rapportgainy);
rapportgainz=rapportgain;rapportgainz(:,:,taille(3))=[];rapportgainz=cat(3,rapportgainz(:,:,1),rapportgainz);
rapportgainx2=rapportgain;rapportgainx2(1,:,:)=[];rapportgainx2=cat(1,rapportgainx2,rapportgainx2(taille(1
)-1,:,:));
rapportgainy2=rapportgain;rapportgainy2(:,1,:)=[];rapportgainy2=cat(2,rapportgainy2,rapportgainy2(:,taille(
2)-1,:));
rapportgainz2=rapportgain;rapportgainz2(:,:,1)=[];rapportgainz2=cat(3,rapportgainz2,rapportgainz2(:,:,taille
(3)-1));
rapportcx=rapportc;rapportcx(taille(1),:,:)=[];rapportcx=cat(1,rapportcx(1,:,:),rapportcx);
rapportcy=rapportc;rapportcy(:,taille(2),:)=[];rapportcy=cat(2,rapportcy(:,1,:),rapportcy);
rapportcz=rapportc;rapportcz(:,:,taille(3))=[];rapportcz=cat(3,rapportcz(:,:,1),rapportcz);
rapportcx2=rapportc;rapportcx2(1,:,:)=[];rapportcx2=cat(1,rapportcx2,rapportcx2(taille(1)-1,:,:));
rapportcy2=rapportc;rapportcy2(:,1,:)=[];rapportcy2=cat(2,rapportcy2,rapportcy2(:,taille(2)-1,:));
rapportcz2=rapportc;rapportcz2(:,:,1)=[];rapportcz2=cat(3,rapportcz2,rapportcz2(:,:,taille(3)-1));

V=V-
((Qmoinsx>0).*rapportgainx.*rapport.*Qmoinsx+(Qmoinsy>0).*rapportgainy.*rapport.*Qmoinsy+(Qmoin
sz>0).*rapportgainz.*rapport.*Qmoinsz+(Qx>0).*rapportgainx2.*rapport.*Qx+(Qy>0).*rapportgainy2.*rap
port.*Qy+(Qz>0).*rapportgainz2.*rapport.*Qz+(Qmoinsx<0).*rapportgain.*rapportx.*Qmoinsx+(Qmoinsy
<0).*rapportgain.*rapporty.*Qmoinsy+(Qmoinsz<0).*rapportgain.*rapportz.*Qmoinsz+(Qx<0).*rapportgai
n.*rapportx2.*Qx+(Qy<0).*rapportgain.*rapporty2.*Qy+(Qz<0).*rapportgain.*rapportz2.*Qz)*echelletem
ps;
C=C-
(pertec.*rapportc+((Qcmoinsx<0).*Qcmoinsx.*rapportcx+(Qcmoinsy<0).*Qcmoinsy.*rapportcy+(Qcmoins
z<0).*Qcmoinsz.*rapportcz+(Qcx<0).*Qcx.*rapportcx2+(Qcy<0).*Qcy.*rapportcy2+(Qcz<0).*Qcz.*rappo
rtcz2)*echelletemps);

v1=sum(V.*~bZ0,3);c1=sum(C.*~bZ0,3);v2=squeeze(sum(V.*~bZ0,2));c2=squeeze(sum(C.*~bZ0,2));
imageeau1=zeros(186,186,3);imageN1=imageeau1;imageeau1(:,:,3)=floor((255/55)*v1);imageN1(:,:,3)=flo
or((255/(((3*400*10^-3)/(3*1.5))/86800)*echelletemps*50)*c1);
eval(['imwrite(imageeau1,"eaudessus1' num2str(date(1)) '.bmp","BMP")'])
eval(['imwrite(imageN1,"nitratedessus1' num2str(date(1)) '.bmp","BMP")'])
imageeau2=zeros(186,43,3);imageN2=imageeau2;imageeau2(:,:,3)=floor((255/55)*v2);imageN2(:,:,3)=floo
r((255/(((3*400*10^-3)/(3*1.5))/86800)*echelletemps*50)*c2);
eval(['imwrite(imageeau2,"eaudessus2' num2str(date(1)) '.bmp","BMP")'])
eval(['imwrite(imageN2,"nitratedessus2' num2str(date(1)) '.bmp","BMP")'])
eval(['save iteration' num2str(date(1)) '.mat v1 c1 v2 c2'])
end
```

Annexe 2 :

source de la croix de bois

site de prélèvement 2

prélèvement site 2

fossé du chemin de la sonnette.

ligne de sources.

ligne de sources.