Société de Calcul Mathématique, S. A.
Algorithmes et Optimisation

Méthode probabiliste pour la détermination de la loi conjointe des débits des rivières

Rapport final

adressé à la

Caisse Centrale de Réassurance

(à l'attention de M. Antoine Quantin)

par la

Société de Calcul Mathématique SA

Le 7 octobre 2009

Rédaction : Sophie Davin, Hélène Bickert

Siège social et bureaux : 111, Faubourg Saint Honoré, 75008 Paris. Tel : 01 42 89 10 89. Fax : 01 42 89 10 69. www.scmsa.com
Société Anonyme au capital de 56 200 Euros. RCS : Paris B 399 991 041. SIRET : 399 991 041 00035. APE : 7219Z

Résumé

La Caisse Centrale de Réassurance (CCR) intervient dans la réassurance des sinistres importants, notamment en cas de catastrophe naturelle (inondations, etc.). Ce sont, par définition, des événements rares et de fortes conséquences.

La CCR souhaite mettre en œuvre une méthode probabiliste globale, portant sur les inondations, et plus précisément sur les débits des cours d'eau. Celui-ci n'est certainement pas suffisant, à lui seul, pour décrire l'étendue et la durée d'une inondation, mais il s'agit d'un critère bien défini, dont les données sont facilement accessibles.

L'objectif de l'étude est de mettre en place la méthode de construction de la loi conjointe des débits des cours d'eau d'une zone, à partir d'un historique de ces débits. Pour ce faire, nous prenons l'exemple concret du bassin versant de la Seine, mais la méthode est transposable à d'autres zones, d'autres risques et d'autres indicateurs.

Nous utilisons les données de débit journalier dont dispose la Banque HYDRO. Le bassin versant de la Seine contient 333 stations de mesure.

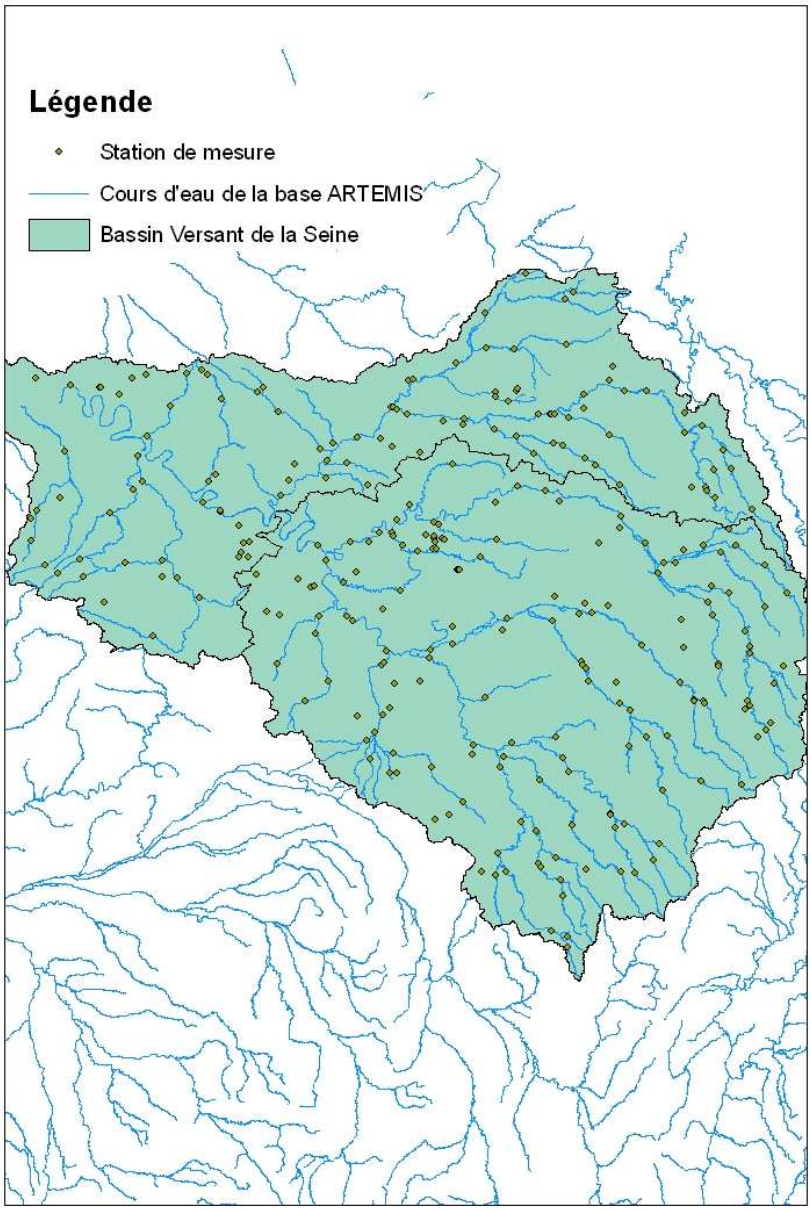


Figure 1 : carte du bassin versant de la Seine issue du logiciel ARCGIS

Extraction des données

Nous avons procédé à l'extraction des données de débit journalier de ces stations (noté QJO), soit sur le site Internet http://hydro.eaufrance.fr/, soit par l'intermédiaire du logiciel HYDRO2 installé dans les locaux de la CCR. Nous avons opté en général pour la seconde solution, moins coûteuse en temps. Certaines stations n'étant pas accessibles via le logiciel, nous les avons récupérées année après année sur le site Internet.

Les données sont exportées en format .txt , et sont nombreuses. Il faut alors les importer sous Excel, extraire les données de débit journalier (QJO) et les mettre sous forme matricielle : chaque colonne représente une station et chaque ligne une date. L'étape de sélection des QJO s'est révélée longue, dans la mesure où les données étaient en nombre trop important pour

contenir dans une seule feuille Excel. Pour réaliser le même traitement sur les données d'autres bassins versants, nous recommandons à la CCR de préférer Access ou tout autre logiciel de traitement de base de données.

Ces étapes d'extraction et de mise en forme des données sont longues ; de nombreuses difficultés ont été rencontrées, notamment avec le logiciel HYDRO2. C'est pourquoi il nous manque les données de 22 stations.

Analyse statistique des données

Nous avons mené une analyse statistique des données acquises, afin de caractériser la quantité de données disponibles. En raison des problèmes rencontrés avec le logiciel HYDRO2, les résultats ne portent que sur les données de 311 stations.

Pour 69 stations, nous ne disposons d'aucune donnée de débit journalier, soit 22% des stations du bassin versant de la Seine. Nous n'exploitons donc les données que de 242 stations.

Le tableau ci-dessous présente les statistiques du nombre et du pourcentage de données manquantes, classées par type de donnée. Il s'agit de statistiques sur toute la période de relevés disponibles.

	QJO
date du 1er relevé	01/01/1936
date du dernier relevé	16/07/2009
nombre de relevés	2 215 377
nombre de relevés possible par station : nombre de jours de la période considérée	26 861
nombre de relevés possibles au total	8 353 771
% de données manquantes	73,48%
nombre de stations sans relevé	69
nombre de relevés possibles en tenant compte des stations relevées uniquement	6 500 362
% de données manquantes en tenant compte des stations relevées uniquement	**65,92%**

Tableau 1 : Analyse statistique des données disponibles

La figure ci-dessous représente le pourcentage moyen de stations relevées par an, pour les données de débit *QJO* :

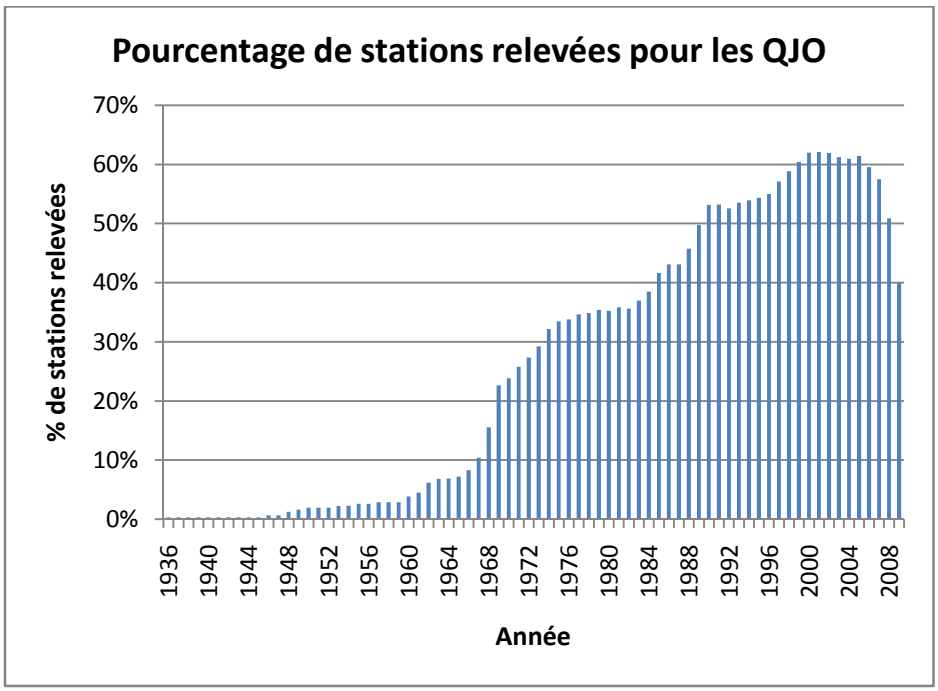


Figure 2 : Pourcentage de stations relevées par année entre 1900 et 2009

La disponibilité des données varie selon les années. Avant les années 1960 – 1970, le nombre de stations relevées est très faible (moins de 5 % des 311 stations considérées). A partir de 1970, on dispose de davantage de données de débit, le maximum étant dans les années 1990.

Nous avons effectué une seconde analyse des données de QJO, en ne prenant en compte que les données à partir de 1970. Les résultats sont regroupés dans le tableau ci-dessous.

	QJO après 1970
date du 1er relevé	01/01/1970
date du dernier relevé	16/07/2009
nombre de relevés	2 074 831
nombre de relevés possible par station : nombre de jours de la période considérée	14 442
nombre de relevés possibles au total	4 491 462
% de données manquantes	53,80%
nombre de stations sans relevé	69
nombre de relevé possibles en tenant compte des stations relevées uniquement	3 494 964
% de données manquantes en tenant compte des stations relevées uniquement	**40,63%**

Tableau 2 : Analyse statistique des données disponibles (après 1970)

On constate que le pourcentage de données manquantes passe de 66% à 41% pour les QJO si on ne considère que les stations relevées après 1970. Ce pourcentage reste tout de même élevé.

Par ailleurs, nous avons étudié les données disponibles et recherché d'éventuelles anomalies parmi elles :

- données manquantes dues à des crues : 14 cours d'eau sont concernés sur 27 dates ou périodes entre septembre et avril. Ce nombre est très faible en regard du nombre total de données. Nous ne n'utiliserons donc pas de méthode de reconstitution particulière.

- débits de valeur nulle suspects : nous avons repéré 881 relevés pour lesquels l'historique indique un débit nul, alors que les 0 affichés correspondent en fait à une absence de relevés. Ces valeurs ont été supprimées de l'historique et sont à reconstituer.

Reconstruction des données

On voit que dans la pratique, les données de débit dont nous disposons pour les rivières sont incomplètes : à certaines dates, la mesure n'a pas été effectuée, a été incorrecte ou n'a pas été enregistrée dans la base de données.

Pour pallier cette absence de relevés, nous avons développé une méthode de reconstitution des données manquantes basée sur la théorie présentée dans le livre [2].

Nous avons implémenté les algorithmes en Matlab, de manière générique : ils peuvent être appliqués à tout type de données, tout nombre de stations et de dates.

Loi conjointe

Nous avons réalisé et implémenté les algorithmes de construction de loi conjointe en Matlab. L'objectif est de construire la loi conjointe de n stations, à partir de l'historique de ces stations constitué de d dates.

De la même façon que pour la reconstruction de données, les programmes créés sont génériques, en ce sens que n'importe quelles données d'entrée peuvent être utilisées : le nombre de stations et de dates sont modifiables. Nous l'appliquons à des données de débit, mais toute autre grandeur peut être exploitée.

Fonctions d'exploitation de la loi conjointe

A la demande de la CCR, nous avons développé un certain nombre de fonctions permettant d'exploiter la loi conjointe :

- calcul de la probabilité d'un événement ;
- tirage d'évènements suivant la loi conjointe ;
- recherche de données correspondant à un événement ;
- calcul des lois marginales ;
- utilisation de probabilités conditionnelles.

Nous appliquons l'ensemble des fonctions codées à un exemple, afin d'illustrer leur utilisation.

Sommaire

I. Problématique

La Caisse Centrale de Réassurance (CCR) intervient dans la réassurance des sinistres importants, notamment en cas de catastrophe naturelle (inondations, etc.). Ce sont, par définition, des événements rares et de fortes conséquences.

L'étendue de chaque événement et sa probabilité vont conditionner la prime d'assurance correspondante, mais ni l'étendue ni la probabilité ne sont faciles à définir. Dans le cas d'une inondation, par exemple, cela signifie : quelle sera la zone couverte, quelle sera la hauteur d'eau, et combien de fois par siècle peut-on attendre un phénomène d'ampleur donnée ?

L'utilisation de la pluviométrie pour répondre à ces questions n'est pas aisée, et est coûteuse en temps. C'est pourquoi la CCR souhaite mettre en œuvre une méthode probabiliste globale, plus robuste que les modèles pluie-débit. Cette méthode porte sur les inondations, et plus précisément sur les débits des rivières. Celui-ci n'est certainement pas suffisant, à lui seul, pour décrire l'étendue et la durée d'une inondation, mais il s'agit d'un critère bien défini, dont les données sont facilement accessibles.

Pour une zone donnée, il s'agit de déterminer la loi conjointe des débits de l'ensemble des cours d'eau de la zone. Elle caractérise leur comportement collectif (par opposition à individuel) et permet de simuler des évènements passés, fictifs, localisés ou bien concernant toute une zone. La CCR a développé le logiciel ARTEMIS qui, à partir de valeurs de débit sur une zone donnée, calcule un coût associé. Elle utilisera donc la loi conjointe des débits pour simuler un grand nombre d'évènements et construire ensuite la loi de probabilité des coûts liés aux risques d'inondation.

L'objectif de l'étude est donc de mettre en place la méthode de construction de la loi conjointe des débits d'une zone. Pour ce faire, nous prenons l'exemple concret du bassin versant de la Seine, mais la méthode est transposable à d'autres zones, d'autres risques et d'autres indicateurs.

Nous utilisons un historique des débits des cours d'eau de ce bassin versant. A partir de cet historique, s'il est complet et d'une ancienneté suffisante, on peut reconstituer la loi conjointe.

Or, dans la pratique, les données de débit dont nous disposons pour les rivières sont incomplètes : à certaines dates, la mesure n'a pas été effectuée, a été incorrecte ou n'a pas été enregistrée dans la base de données.

Pour pallier cette absence de relevés, nous avons développé une méthode de reconstitution des données manquantes basée sur la théorie présentée dans le livre [2].

II. Données disponibles

A. Nature des données

Nous étudions le bassin versant de la Seine. Sur chacun de ses affluents et sur la Seine, des stations de mesures récoltent des données de hauteur d'eau.

Ces mesures sont effectuées par différents services de l'État comme les DIREN, les Agences de l'Eau, les Services de Prévision des Crues, le Cemagref… Chaque producteur de données vérifie et valide ses mesures selon sa propre appréciation, puis alimente la « Banque HYDRO ».

HYDRO calcule sur une station donnée les débits instantanés, journaliers, mensuels... à partir des valeurs de hauteur d'eau et des courbes de tarage (relations entre les hauteurs et les débits). Ces valeurs sont actualisées à chaque mise à jour d'une hauteur ou d'une courbe de tarage (addition, précision supplémentaire, correction...).

Nous avons ainsi accès à quatre types de données :

- HMM : hauteur d'eau maximale instantanée mensuelle (en mètres et centimètres) ;
- QJO : débit journalier correspondant à la moyenne des débits sur une journée donnée (en l/s et m^3/s) ;
- QME : débit mensuel observé (en l/s et m^3/s) ;
- QMM : débit mensuel maximum (en l/s et m^3/s) ;

Nous nous intéressons principalement aux QJO car ces données journalières permettent d'identifier des évènements extrêmes.

B. Stations de mesure

Le bassin versant de la Seine contient 333 stations de mesure. Parmi elles, nous distinguons deux types de stations :

- Les stations situées sur des cours d'eau de la base ARTEMIS. Il s'agit de la liste des principaux cours d'eau, déterminée par la CCR : 220 stations sont ainsi concernées ;

- Les stations situées sur des cours d'eau hors ARTEMIS : il s'agit de 113 stations.

Dans un premier temps, il a été question de se limiter aux cours d'eau de la base ARTEMIS. Cette décision a été modifiée ; nous avons alors intégré la totalité des stations relevées à notre étude.

Pour représenter les cours d'eau et les stations, nous utilisons le logiciel de cartographie ARCGIS : il permet de superposer différentes bases de données et formes géographiques sur une même carte.

Dans notre cas nous avons superposé trois fichiers :

- la base HYDRO (emplacement des 333 stations du bassin versant de la Seine) ;
- la base ARTEMIS ;
- les contours du bassin versant de la Seine.

La figure ci-après représente la superposition par ARCGIS de ces trois fichiers. Les stations sont représentées par des points (en vert sur la carte), les cours d'eau sont représentés par des tronçons (en bleu sur la carte), et le bassin versant de la Seine est représenté par la forme coloriée en bleu.

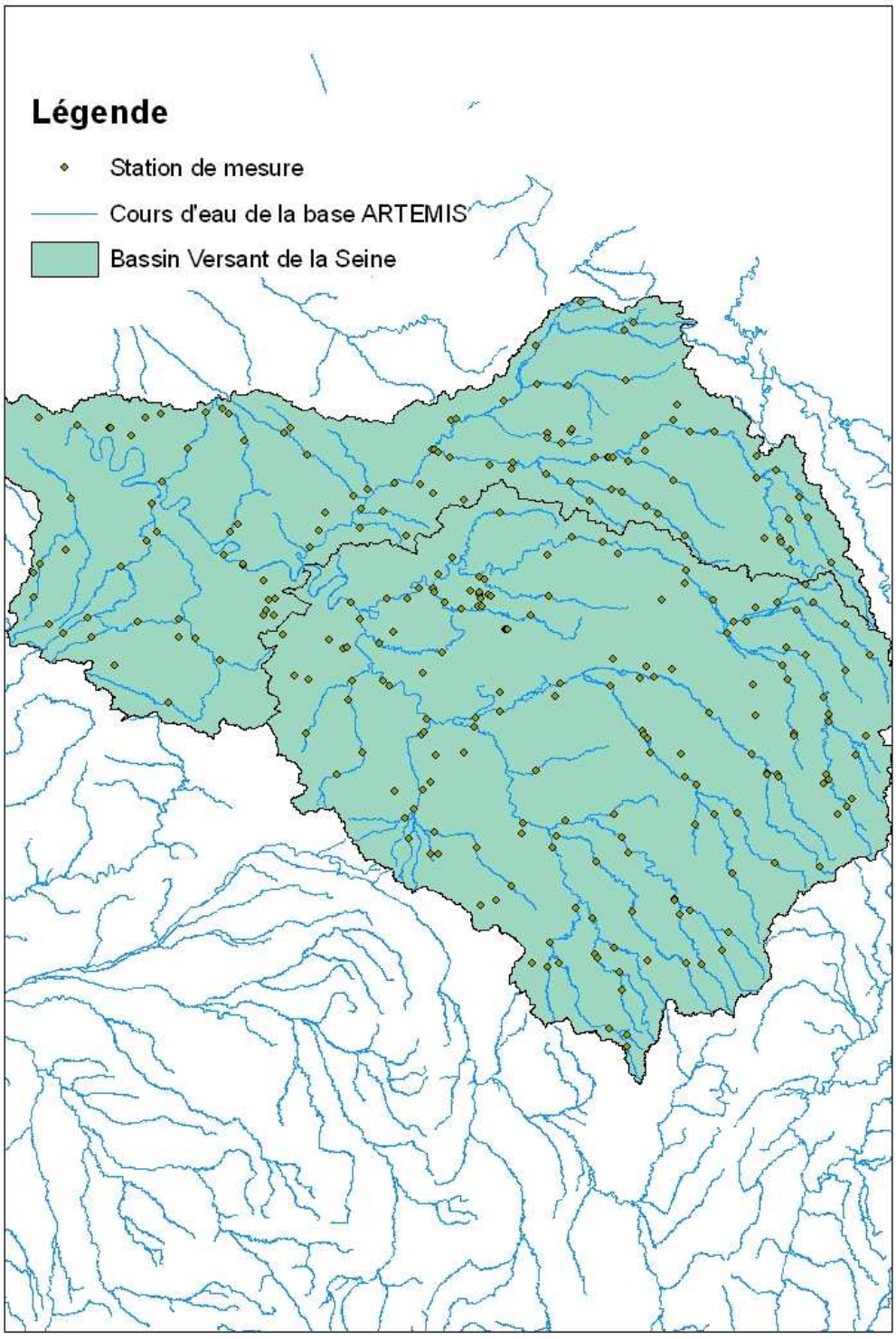


Figure 3 : carte du bassin versant de la Seine issue du logiciel ARCGIS

ARGIS permet de faire des recoupements entre les différents fichiers d'entrée. Par exemple, il est possible d'extraire l'ensemble des stations qui intersectent les tronçons. Nous avons utilisé cette fonctionnalité pour construire la liste des stations situées sur des cours d'eau de la base ARTEMIS.

En outre, les bases HYDRO et ARTEMIS ne sont pas parfaitement superposées et certains points ont été exclus à tort. Nous avons donc vérifié manuellement les points litigieux et nous avons ajouté un certain nombre de stations. Nous avons au final 220 points de mesure parmi les 333 initiaux.

La figure ci-dessous représente ces 220 stations.

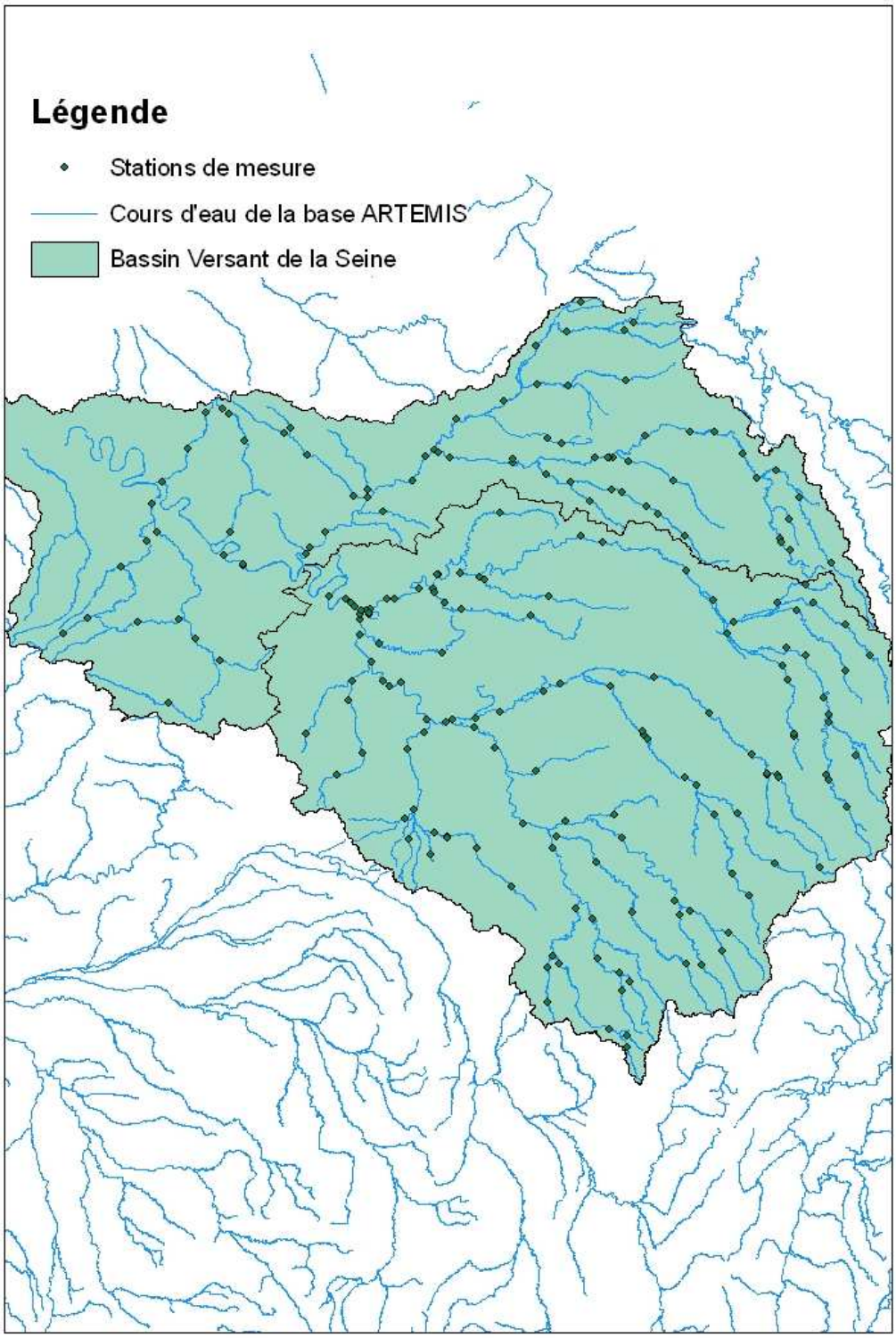


Figure 4 : carte du bassin versant de la Seine issue du logiciel ARCGIS

C. Extraction des données

Nous avons procédé à l'extraction des données de débit (QJO).

Les données de la Banque HYDRO peuvent être récupérées de deux manières : soit sur le site Internet http://hydro.eaufrance.fr/, soit par l'intermédiaire du logiciel HYDRO2 installé dans les locaux de la CCR.

En fonction des stations à récupérer, nous avons utilisé l'une ou l'autre des méthodes.

Il n'est pas possible d'extraire toutes les années de toutes les stations de mesures en même temps. Sur le site Internet, on peut extraire jusqu'à cent stations à la fois mais il faut procéder année par année. Inversement, avec le logiciel, on peut extraire toutes les années d'un coup mais il faut procéder par « paquets » d'une dizaine de stations.

Nous avons opté en général pour la seconde solution, moins coûteuse en temps. Certaines stations n'étant pas accessibles via le logiciel, nous les avons récupérées année après année sur le site Internet.

Les données sont exportées en format .txt : pour chaque station, les données de HMM, QJO, QME et QMM apparaissent les unes à la suite des autres. L'étape suivante consiste à les importer sous Excel, extraire les données de débit journalier (QJO) et les mettre sous forme matricielle : chaque colonne représente une station et chaque ligne une date. L'étape de sélection des QJO s'est révélée longue, dans la mesure où les données étaient en nombre trop important pour contenir dans une seule feuille Excel.

Pour réaliser le même traitement sur les données d'autres bassins versants, nous recommandons à la CCR de préférer Access ou tout autre logiciel de traitement de base de données.

Difficultés rencontrées

L'étape d'acquisition des données est longue et laborieuse. Comme on l'a dit, il n'est pas possible d'extraire l'historique des 333 stations simultanément : nous avons dû l'extraire par dizaines (voire moins). Ceci représente de nombreuses heures de travail.

De plus, il arrive que l'export des données échoue, pour des raisons inconnues (bugs du logiciel) ; il faut alors effectuer une seconde fois toutes les étapes de préparation des données. Lorsque l'export n'échoue pas, il arrive que le logiciel « coupe » les données à partir d'une certaine taille de fichier : lorsque le nombre de données est trop important, il n'exporte qu'une certaine quantité de données. Aucun message d'erreur ne le signale ; il faut donc vérifier les fichiers de tous les exports.

En outre, certaines stations ne sont pas accessibles via le logiciel. Pour celles-ci, nous avons dû utiliser le site Internet : l'export des données y est beaucoup plus long que par le logiciel (site lent à charger, nombreuses étapes de préparation des exports).

Pour d'autres stations, aucune donnée n'est disponible.

Par ailleurs, après un certain nombre d'exports, il n'a plus été possible d'exporter quoi que ce soit. C'est pourquoi il nous manque encore les données d'une vingtaine de stations. Nous ignorons d'où provient ce problème.

Enfin, l'unité des données extraites nous est toujours inconnue (l/s ou m^3/s).

Le 28 juillet 2009, la CCR et la SCM ont adressé une liste de questions au SCHAPI (Service Central d'Hydrométéorologie et d'Appui à la Prévision des Inondations), en charge des problèmes rencontrés avec les données de la banque HYDRO. A ce jour, nous n'avons pas obtenu de réponse.

D. Analyse statistique des données

L'objectif de ce paragraphe est de qualifier la disponibilité des relevés. En raison des problèmes rencontrés avec le logiciel HYDRO2, nous ne disposons des données que de 311 stations. Les conclusions de cette partie seront donc modifiées lorsque nous disposerons de l'ensemble des données (333 stations).

a. Statistiques globales

Le tableau page suivante présente les statistiques du nombre et du pourcentage de données manquantes, classées par type de donnée.

Il s'agit de statistiques sur toute la période de relevés disponibles, incluant les premières années qui ne contiennent que très peu de données.

	QJO
date du 1er relevé	01/01/1936
date du dernier relevé	16/07/2009
nombre de relevés	2 215 377
nombre de relevés possible par station : nombre de jours de la période considérée	26 861
nombre de relevés possibles au total	8 353 771
% de données manquantes	73,48%
nombre de stations sans relevé	69
nombre de relevés possibles en tenant compte des stations relevées uniquement	6 500 362
% de données manquantes en tenant compte des stations relevées uniquement	**65,92%**

Tableau 3 : Analyse statistique des données disponibles

Le pourcentage de données manquantes est très élevé parmi ces 311 stations. Cela est en partie dû à un très faible nombre de relevés les premières décennies : en effet, la date de 1er relevé (1936) ne concerne qu'une minorité de stations. La plupart ont été relevées à partir des années 1960 – 1970 (voir *figure 3* dans la suite du rapport).

Nous avons regardé pour les QJO en ne prenant en compte que les données à partir de 1970. Les résultats sont regroupés dans le tableau page suivante.

	QJO après 1970
date du 1er relevé	01/01/1970
date du dernier relevé	16/07/2009
nombre de relevés	2 074 831
nombre de relevés possible par station : nombre de jours de la période considérée	14 442
nombre de relevés possibles au total	4 491 462
% de données manquantes	53,80%
nombre de stations sans relevé	69
nombre de relevé possibles en tenant compte des stations relevées uniquement	3 494 964
% de données manquantes en tenant compte des stations relevées uniquement	**40,63%**

Tableau 4 : Analyse statistique des données disponibles (après 1970)

On constate que le pourcentage de données manquantes passe de 66% à 41% pour les QJO si on ne considère que les stations relevées après 1970. Ce pourcentage reste tout de même élevé.

b. Stations non relevées

Pour 69 stations, nous n'avons aucune donnée de débit journalier, soit 22% des stations du bassin versant de la Seine. Nous n'exploitons donc les données que de 242 stations.

La carte ci-dessous localise ces stations.

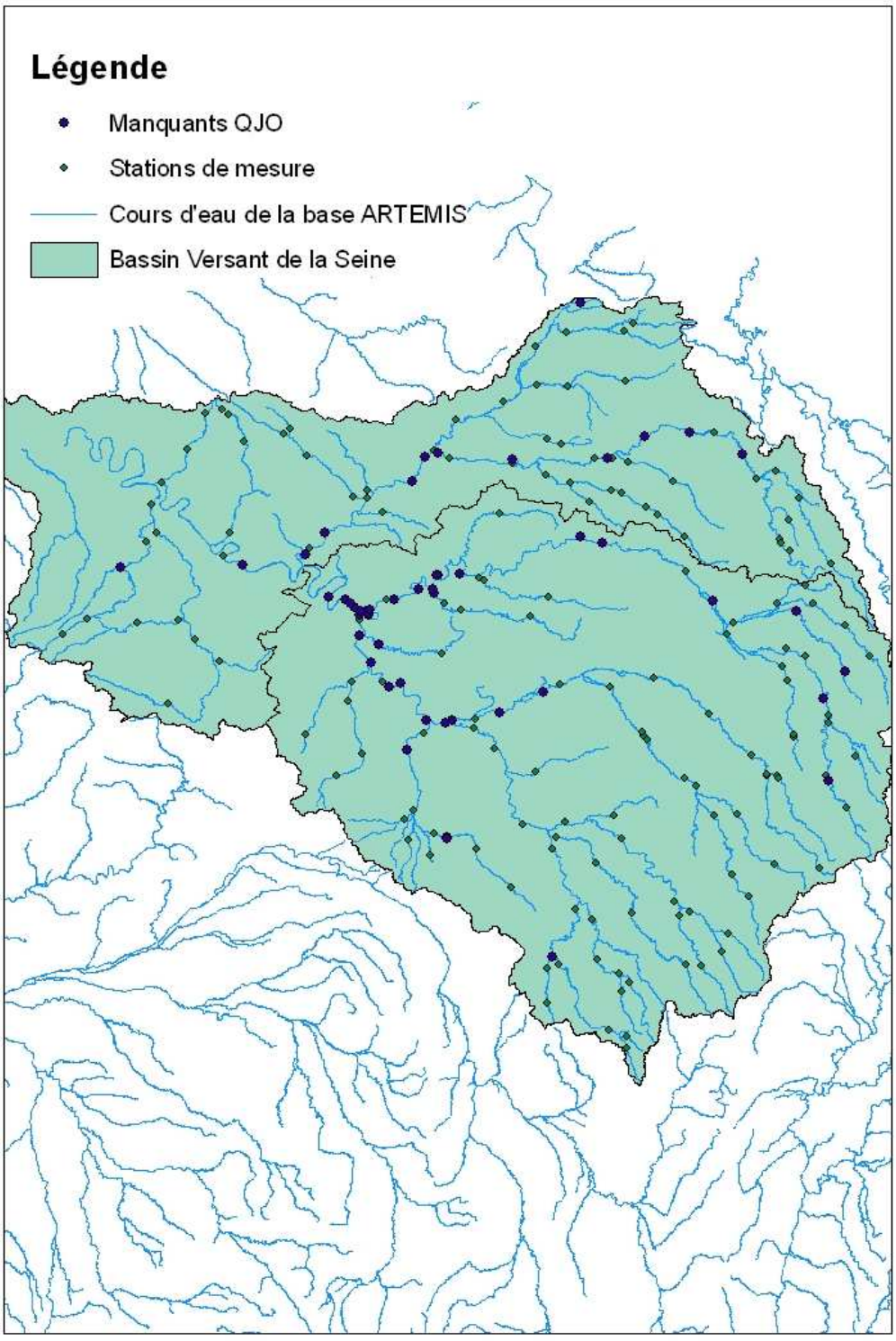


Figure 5 : carte du bassin versant de la Seine issue du logiciel ARCGIS

Les stations sur lesquelles aucun relevé de QJO n'a été effectué sont faiblement dispersées ; on constate une forte concentration aux abords de l'Ile-de-France.

c. *Disponibilité annuelle*

La figure ci-dessous représente le pourcentage moyen de stations relevées par an, pour les données de débit *QJO* :

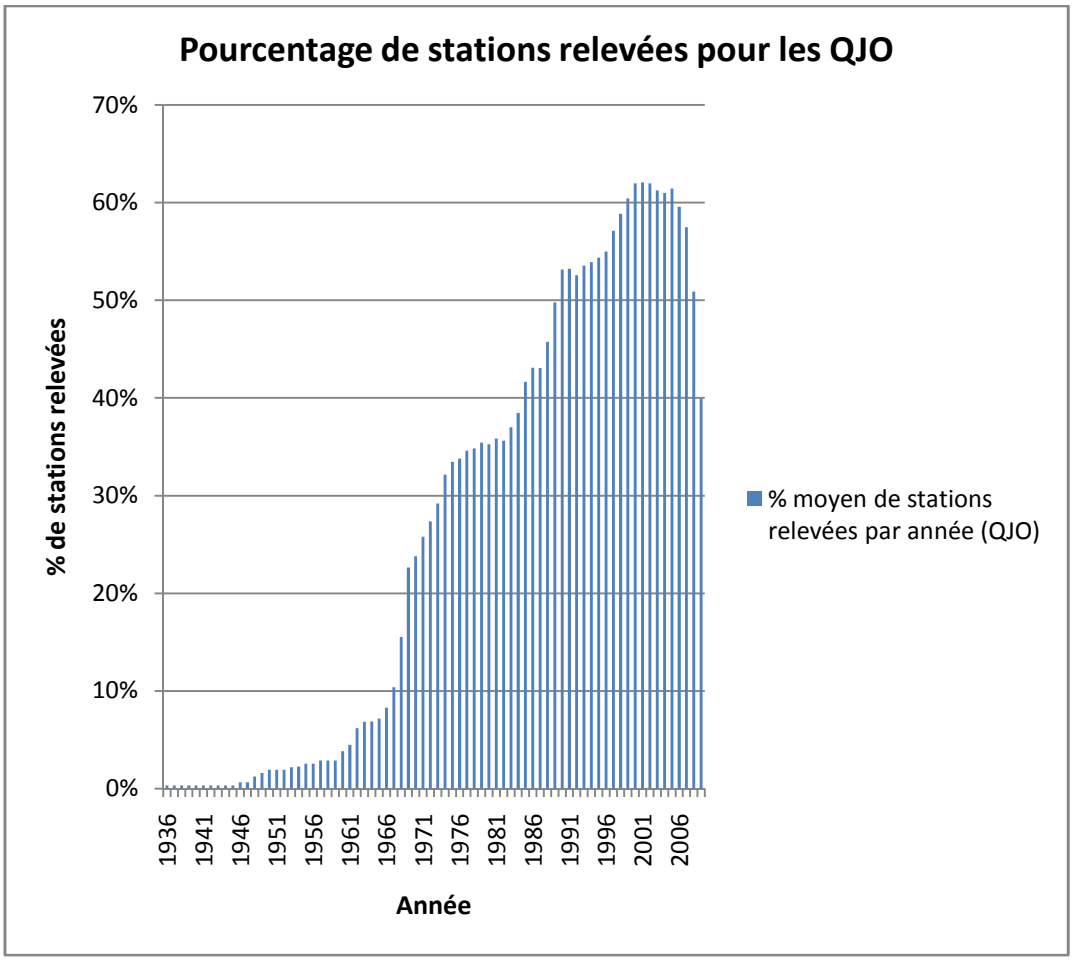


Figure 6 : Pourcentage de stations relevées par année entre 1900 et 2009

La disponibilité des données varie selon les années. Avant les années 1960 – 1970, le nombre de stations relevées est très faible (moins de 5 % des 311 stations considérées). A partir de 1970, on dispose de davantage de données de débit, le maximum étant dans les années 1990.

d. *Répartition du nombre de stations en fonction du nombre de mesures*

Ci-dessous, nous représentons la répartition du nombre de stations en fonction du nombre de mesures effectuées entre la date du premier relevé et celle du dernier relevé.

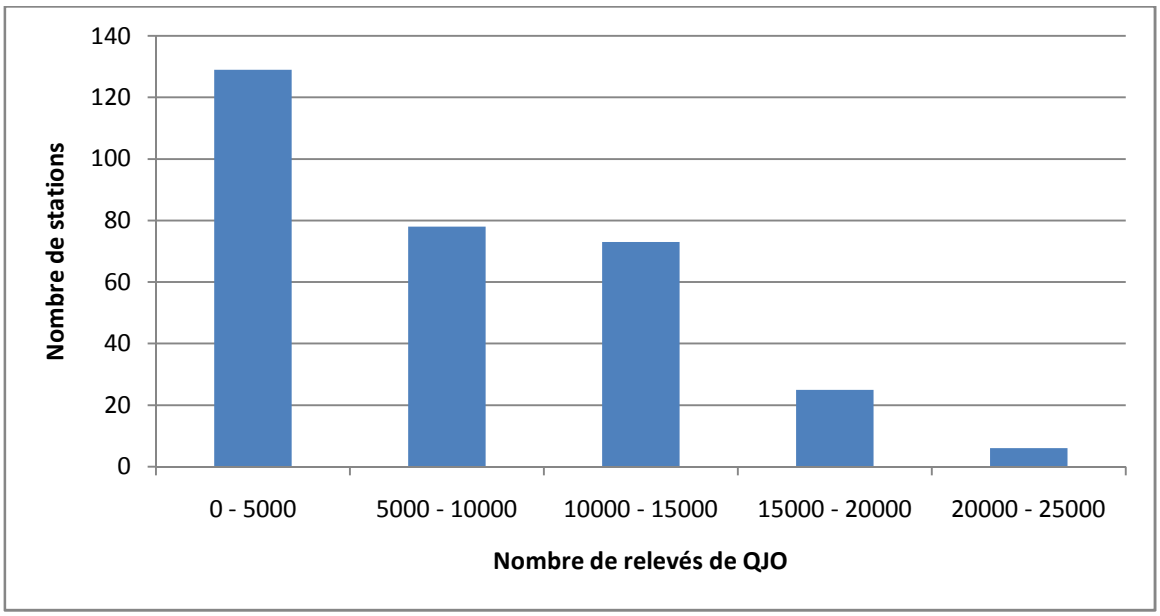


Figure 7 : Répartition du nombre de stations en fonction du nombre de relevés de QJO

On constate près de la moitié des stations ont été relevées sur moins de 15% de l'historique : cela signifie que, pour ces stations, le pourcentage de données manquantes est au moins de 85%.

Plus précisément, la répartition est la suivante pour les mesures de débit journalier (QJO) :

- 41% des stations ont entre 81% et 100% de données manquantes ;
- 49% des stations ont entre 44% et 81% de données manquantes ;
- Seulement 10% des stations ont moins de 44% de données manquantes.

Ces chiffres sont très élevés. Cependant, ils sont liés à la date du premier relevé sur l'ensemble des stations. On a vu dans la figure 3 de ce rapport, qu'entre 1900 et les années 60 moins de 5% des stations ont été relevées ; c'est ce qui engendre ces pourcentages de données manquantes aussi importants.

E. Analyse statistique des stations non ARTEMIS

Nous avons réalisé le même traitement sur les stations non ARTEMIS uniquement : 113 stations sont concernées. Les 22 stations dont les données nous manquent étant des stations hors ARTEMIS, l'étude ci-dessous ne porte que sur les 91 stations restantes.

L'objectif est de déterminer si, parmi ces stations, certaines ont un historique important, et mériteraient ainsi d'être ajoutée à la base ARTEMIS de la CCR.

a. Statistiques globales

Le tableau ci-dessous regroupe les statistiques globales pour l'ensemble des données des stations non ARTEMIS :

	QJO
Date du 1er relevé	01/01/1961
Date du dernier relevé	16/07/2009
Nombre de relevés	697 455
Nombre de relevés possibles par station (nombre de jour ou de mois)	17 729
Nombre de relevés possibles au total (nb stations * nb relevés possibles par station)	1 613 339
% de données manquantes	56,77 %
Nombre de stations sans relevé	8
Nombre de relevés possibles en tenant compte des stations relevées uniquement	1 471 507
% de données manquantes en tenant compte des stations relevées uniquement	**52,60 %**

Tableau 5 : Analyse statistique des données disponibles des stations hors ARTEMIS

Dans ce tableau, les dates de relevées ont été arrêtées au 16/07/2009.

On constate que le pourcentage de données manquantes est du même ordre que pour les stations ARTEMIS.

Nous avons recalculé ces statistiques en ne prenant que les QJO mesurés après 1970 :

	QJO après 1970
Date du 1er relevé	01/01/1970
Date du dernier relevé	16/07/2009
Nombre de relevés	674 039
Nombre de relevés possibles par station : nombre de jours sur la période	14 441
Nombre de relevés possibles au total : nombre de stations * nb relevés possibles par station	1 314 131
% de données manquantes	48,71 %
Nombre de stations sans relevé	8
Nombre de relevés possibles en tenant compte des stations relevées uniquement	1 198 603
% de données manquantes en tenant compte des stations relevées uniquement	**43,76 %**

Tableau 6 : Analyse statistique des données disponibles des stations hors ARTEMIS, après 1970

b. Stations bien relevées

Nous souhaitons savoir s'il y a des stations bien relevées parmi les stations non ARTEMIS. Pour cela, nous avons tracé la fonction de répartition du pourcentage de données manquantes pour QJO. On constate qu'elle est linéaire :

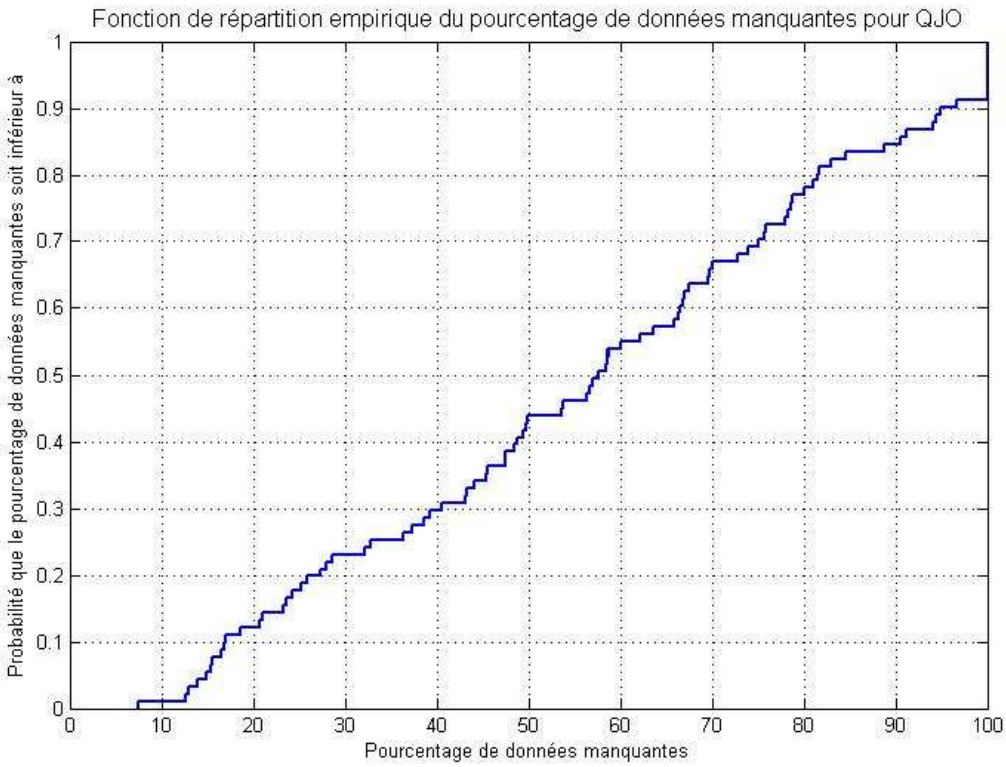


Figure 8 : Fonction de répartition du pourcentage de données QJO manquantes

Cette courbe se lit de la manière suivante : prenons un pourcentage de données manquantes de 20 % (en abscisse) ; 12 % des 91 stations non ARTEMIS ont un pourcentage de données manquantes inférieur à 20, soient 11 stations. Il s'agit des stations suivantes :

Station	nombre de relevés (sur 17729)	date du 1er relevé	date du dernier relevé	données manquantes dans la période de relevé	% de données manquantes dans la période de relevés	nbre de données manquantes sur l'historique commun	% de données manquantes sur l'historique commun
H2073110	15 494	31/01/1967	02/07/2009	0	0,00 %	2 235	12,61 %
H2083110	14 729	01/01/1969	30/06/2009	62	0,42 %	3 000	16,92 %
H2513110	14 988	13/06/1968	02/07/2009	7	0,05 %	2 741	15,46 %
H3613010	15 421	01/01/1963	02/07/2009	1 564	9,21 %	2 308	13,02 %
H4223110	15 098	29/02/1968	16/07/2009	16	0,11 %	2 631	14,84 %
H4243010	14 439	01/01/1968	16/07/2009	734	4,84 %	3 290	18,56 %
H4252010	15 263	28/09/1967	16/07/2009	5	0,03 %	2 466	13,91 %
H5302010	16 405	01/01/1961	02/07/2009	1 310	7,39 %	1 324	7,47 %
H7423710	14 970	15/07/1968	09/07/2009	0	0,00 %	2 759	15,56 %
H7513010	14 720	15/07/1968	09/07/2009	250	1,67 %	3 009	16,97 %
H7853010	14 810	29/12/1968	16/07/2009	0	0,00 %	2 919	16,46 %

Tableau 7 : Stations hors ARTEMIS dont le pourcentage de données manquantes est inférieur à 10%

Ces 11 stations peuvent être considérées comme bien représentées du point de vue des QJO ; les cours d'eau qu'elles représentent pourront être intégrés à la base ARTEMIS.

F. Recherche d'anomalies dans les données

Nous avons étudié les données disponibles et recherché d'éventuelles anomalies parmi elles :

- données manquantes dues à des crues ;
- débits de valeur nulle suspects.

a. Relation éventuelle entre absence de relevé et débit élevé

Lorsqu'une rivière a un débit très élevé, il est difficile d'effectuer des jaugeages. Il arrive que dans ces cas-là, la mesure soit impossible.

Nous avons regardé si de tels cas de figure existaient parmi les données dont nous disposons : est-ce que certaines données manquantes surviennent après un épisode de débit particulièrement important ? Si c'est le cas, une méthode particulière de reconstructions de données manquantes « extrêmes » est recommandée (voir [1]).

Nous avons listé les absences de relevés faisant suite à des débits exceptionnellement élevés : 14 cours d'eau sont concernés sur 27 dates ou périodes entre septembre et avril.

Parmi les 27 relevés manquants, 4 correspondent à un 31 décembre et il s'agit plus vraisemblablement d'une absence de l'opérateur. Les 23 autres peuvent être dus à des fortes crues. Ce nombre est très faible en regard du nombre total de données. Nous ne n'utiliserons donc pas de méthode de reconstitution particulière.

b. Recherche de données manquantes cachées

L'historique dont nous disposons contient un certain nombre de valeurs de débit nulles. En observant dans le détail, nous nous sommes aperçus qu'elles pouvaient être de deux types :

- Soit le débit est réellement nul : il n'y a plus d'eau dans la rivière ;

- Soit le 0 affiché est dû à une absence de relevés. Dans ce cas, il s'agit de données manquantes cachées, qu'il faut considérer comme telles pour les étapes de reconstitution de données manquantes et construction de la loi conjointe.

Nous avons recherché ces anomalies dans nos données. Nous avons ainsi dressé deux listes :

- Le 0 affiché correspond à coup sûr à une donnée manquante. Ces valeurs ont été supprimées de l'historique et sont à reconstituer. Environ 700 relevés ont ainsi été modifiés ;

- Le 0 affiché correspond peut-être à une donnée manquante. Dans le doute, nous n'avons pas modifié ces données.

Les deux listes sont données en annexe.

L'impact de la suppression des 0 sur les statistiques globales est insignifiant : le pourcentage de données manquantes passe de 65.93% à 65.92%.

III. Loi conjointe

Nous avons réalisé et implémenté les algorithmes de construction de loi conjointe en Matlab. L'objectif est de construire la loi conjointe de n stations, à partir de l'historique de ces stations constitué de d dates.

Les programmes créés sont génériques, en ce sens que n'importe quelles données d'entrée peuvent être utilisées : le nombre de stations et de dates sont modifiables. Nous l'appliquons à des données de débit, mais toute autre grandeur peut être exploitée.

A. Algorithme de la loi conjointe

a. Allure générale de l'algorithme

Afin de calculer une loi conjointe de débits sur n stations de mesures, on commence par découper, pour chaque station, la plage de valeurs du débit de la station en k intervalles.

Un événement est un n-uplet (ou un vecteur de taille n) dont chaque composante est un entier de 1 à k. Ainsi le nombre d'événements possibles est k^n. Calculer la loi conjointe, c'est calculer la probabilité de chacun de ces événements.

Par exemple, pour 330 stations et 10 intervalles, le nombre d'événements observable est 10^{330}, donc 10^{330} probabilités à calculer.

Dans la suite, nous emploierons parfois le terme classe pour désigner l'intervalle dans laquelle se trouve une valeur de débit.

Nous considérons une période d'étude de d jours, et nous supposons que les n stations ont été relevées tous les jours : cela signifie que nous n'avons pas de données manquantes.

Le nombre d'événements observables dépend entièrement des nombres de stations et de classes considérés.

Le nombre d'événements observés est au maximum d. Ce cas correspondant à des événements distincts tous les jours de l'étude.

Remarquons que le nombre d'événements observables est une fonction exponentielle du nombre de stations, et croît donc très rapidement quand on rajoute des stations pour le calcul de la loi conjointe. En revanche, le nombre d'événements observés ne dépend que du nombre de dates de la période d'étude.

L'algorithme ne va pas considérer chaque événement et compter son nombre d'occurrences, il va se limiter aux cas des événements effectivement observés. Ainsi, de k^n probabilités à calculer, on passe à d probabilités au maximum, ce qui est plus satisfaisant en termes de temps de calcul lorsque le nombre de stations considérées est important.

Supposons par exemple que nous ayons 300 stations, 10 classes et 36500 jours de relevés, soit environ 100 ans. Nous avons pu observer au maximum 36500 événements différents mais le nombre de n-uplets possibles est 10^{300}.

L'algorithme que nous utilisons est :

```
Pour chaque date d'observation :

    Regarder quel événement se produit

    Si l'événement a déjà été observé à une date antérieure, incrémen-
    ter d'une unité le compteur correspondant

    Si c'est la première fois que l'on observe l'événement, créer un
    compteur associé initialisé à 1.

Diviser le nombre d'occurrences de chaque station par le nombre de dates.
```

Une étape préliminaire à cet algorithme consiste à remplacer les données de débit par la classe correspondante.

b. Mise en forme préliminaire des données

Nous disposons dans un fichier Excel des matrices :

- $donnees = (donnees)_{i,j}$: Elle contient les données de débits et se présente sous la forme :

	N° de la $1^{ère}$ station			N° de la i^{e} station			N° de la dernière station
$1^{ère}$ date				|			
				|			
				|			
j^{e} date	_____	_____	_____	Débit de la i^{e} station enregistré à la date j			
Dernière date							

Tableau 8 : Présentation de la feuille Excel contenant les données de débits

La première colonne correspond aux dates, la première ligne aux stations.

La case $(i+1\,,\ j+1)$ contient le débit de la station i enregistré à la date j.

REMARQUES :

- Cette matrice doit être entrée dans le premier onglet du fichier Excel.
- Pour le bon fonctionnement du programme, afin d'éviter des problèmes d'importation entre Matlab et Excel, les débits, les dates et les numéros de stations doivent être au format numérique et non au format texte. Il est donc indispensable de repérer les stations par un numéro et non pas leur nom.
- Le nombre de stations et de dates n'est pas limité a priori.
- Le tableau peut comporter des données manquantes, sachant que si tel est le cas, les dates correspondantes ne seront pas prises en compte dans le calcul des probabilités.

- $interv = (interv)_{i,j}$: Elle contient dans chaque colonne les bornes des intervalles intervenant dans la construction de la loi conjointe.

  Supposons par exemple que pour la station i, nous voulions connaître la probabilité des intervalles :

  $$[a_i, b_i\,[\,, [b_i, c_i\,[\,, [c_i, d_i] \ \text{ avec } a_i < b_i < c_i < d_i$$

  La colonne associée à la station i présentera les valeurs : $a_i, b_i, c_i\ et\ di$

N° de la 1ère station			*N° de la ie station*			*N° de la ne station*
a_1			a_i			a_n
b_1			b_i			b_n
c_1			c_i			c_n
d_1			d_i			d_n

Tableau 9 : Présentation de la feuille Excel contenant les données des intervalles de valeurs de débits

Le nombre d'intervalles ne doit pas nécessairement être le même pour toutes les stations.

REMARQUES :

- Cette matrice doit être rentrée dans le second onglet du fichier Excel.

- Les noms de stations doivent être au format numérique, comme sur la première feuille.

- Le nombre de stations et d'intervalles n'est pas limité a priori.

- Le nombre d'intervalles n'a pas à être identique pour toutes les stations.

- Les valeurs de chaque colonne doivent être rangées dans l'ordre croissant. Dans le tableau ci-dessus il ne faut pas avoir $a_i > b_i$ par exemple.

- Le tableau **ne doit pas** présenter de données manquantes.

c. Implémentation de l'algorithme sous Matlab

Nous commençons par présenter les tests effectués sur les données afin de vérifier leur cohérence.

- Fonction supprimant les lignes du tableau des débits auxquelles il manque une valeur : `suppr_lignes.m`

Pour utiliser cette fonction, un test est d'abord effectué sur chaque ligne du tableau de valeurs des débits afin de détecter s'il manque des données. Si tel est le cas, la fonction `suppr_lignes.m` est appelée, et la ligne en question supprimée.

```
function new=suppr_lignes(tableau,nb_lignes,nb_colonnes,num_ligne)

for i=num_ligne:nb_lignes-1
    tableau(i,1:nb_colonnes)=tableau(i+1,1:nb_colonnes);
end

new=tableau;
```

- Vérification de la cohérence des noms de stations : `verif_stations.m`

Cette fonction a pour but de vérifier que les stations renseignées dans le tableau des débits et celles du tableau des intervalles sont identiques et données dans le même ordre. Si ce n'est pas le cas, un message d'erreur est affiché et le programme arrêté.

```
function v = verif_stations(tab_debits,tab_intervalles,dim)

v=0;
j=0;


for j=1:dim
    if tab_debits(1,j+1)-tab_intervalles(1,j)==0
        v=0;
    else v=v+1;
    end
end
```

- Vérification de l'ordre des bornes des intervalles : `verif_bornes.m`

Les bornes des intervalles renseignés dans le deuxième onglet du classeur Excel doivent nécessairement être données dans l'ordre croissant. Cette fonction s'assure que cette condition est respectée.

```
function ver_int=verif_bornes(tab_intervalles,nb_lignes,nb_colonnes)

i=0;
j=0;
ver_int(1:nb_colonnes)=0;

for j=1:nb_colonnes
    for i=1:(nb_lignes(2,j)-1)
        a=tab_intervalles(i+1,j)-tab_intervalles(i,j);
        if a<=0
            ver_int(j)=ver_int(j)+1;
        else ver_int(j)=ver_int(j)+0;
        end
    end
end
```

- Fonction vérifiant que les valeurs des débits sont incluses dans les intervalles donnés : `verif_debit.m`

Cette fonction compare le débit minimum observé pour chaque station avec la borne inférieure des intervalles et le débit maximum avec la borne supérieure. Si l'un des débits est en dehors des bornes, un message d'erreur est affiché donnant le numéro de la station où le problème a été détecté.

```
function verif=verif_debit(tab_debits,tab_intervalles,k,bornes)

verif=0;
sup=max(tab_debits,[],1);
inf=min(tab_debits,[],1);

for j=1:k
    if sup(j)-tab_intervalles(bornes(2,j),j)<0
        verif(j)=0;
    else verif(j)=1;
    end
end

for j=1:k
    if inf(j)-tab_intervalles(1,j)>=0
        verif(j)=verif(j);
    else verif(j)=verif(j)+1;
    end
end
```

Nous détaillons à présent les deux étapes du calcul de la loi conjointe.

Etape 1 : Remplacement des données de débit par les valeurs de classe

a) Présentation de la fonction `num_inter.m`

Pour cette étape, nous avons besoin d'une fonction qui à partir d'une valeur x et d'une liste d'intervalles, donne l'intervalle dans lequel se trouve x.

Nous avons créé pour effectuer ce calcul la fonction `num_inter.m`.

```
function y=num_inter(x,vect)

y=0;
t=length(vect);

for i=1:t-1
    y=y+( x>=vect(i) );
end
```

En entrée de la fonction, nous considérons une valeur x et un vecteur $vect$ à composantes croissantes. Les composantes de $vect$ représentent les bornes des intervalles considérés.

Nous initialisons un compteur y à 0. Pour chaque composante $vect(i)$ de $vect$, de la première à l'avant dernière, si $x > vect(i)$, nous incrémentons le compteur d'une unité.

Par exemple, si $x_1 = 5{,}5$, $x_2 = 1$, $x_3 = 12$ et $vect = (3,5,7,8,10)$:

- x_1 est supérieur aux deux premières composantes de *vect* donc le résultat de la fonction est 2.
- x_2 est inférieure à toutes les composantes de *vect* donc le résultat de la fonction vaut 0.
- x_3 est supérieur à toutes les composantes de *vect*, c'est-à-dire 5, donc en particulier aux 4 premières. Le résultat obtenu est 4.

Toutes les valeurs supérieures à l'avant dernière composante de *vect* sont considérées comme faisant partie de la même classe. Etant donné que la fonction décrite précédemment, `verif_debit.m`, vérifie que tous les débits mesurés pour une station sont inférieurs à la dernière borne donnée pour les intervalles, ce choix est pertinent.

b) Programme Matlab transformant les données en classes

Voici le programme, utilisant la fonction `num_inter`, remplaçant les données de débits par les classes correspondantes.

```
interv=xlsread('donnees_reconstituees.xls','Feuil2');

donnees=donnees(2:nb_dates+1,2:nb_stations+1);
interv=interv(2:dim(1)+1,1:nb_stations);

M_histo=donnees;

for s=1:nb_stations

      M_histo(1:nb_dates,s)=num_inter(donnees(1:nb_dates,s),interv(1:nb_bornes(2
,s),s));

end
```

Les données des intervalles sont importées avec la fonction `xlsread` depuis le deuxième onglet de la feuille Excel ; elles sont stockées dans la matrice *interv*.

Le choix a été fait de stocker dans des vecteurs les références des stations et les dates, dans le but de s'en affranchir ensuite pour les tableaux des débits et des intervalles, ce qui permet de simplifier les indices utilisés dans les différentes étapes de calcul.

Exemple : pour la matrice M_histo, les indices de ses composantes sont désormais contenus entre 1 et nb_dates, et 1 et $nb\_stations$; au lieu de 2 et $nb\_dates + 1$ et 2 et $nb\_stations + 1$.

La matrice *nb_bornes* stocke le nombre de bornes, renseignés pour chaque station. Ce nombre pouvant varier d'une station à l'autre, ils sont tous conservés en mémoire.

On crée ensuite la matrice *M_histo*, initialisée comme étant égale à la matrice *donnees*.

Toutes les colonnes (stations) de *M_histo* sont parcourues ; les valeurs de débit sont remplacées par les classes correspondantes.

La matrice *M_histo* est donc de la forme :

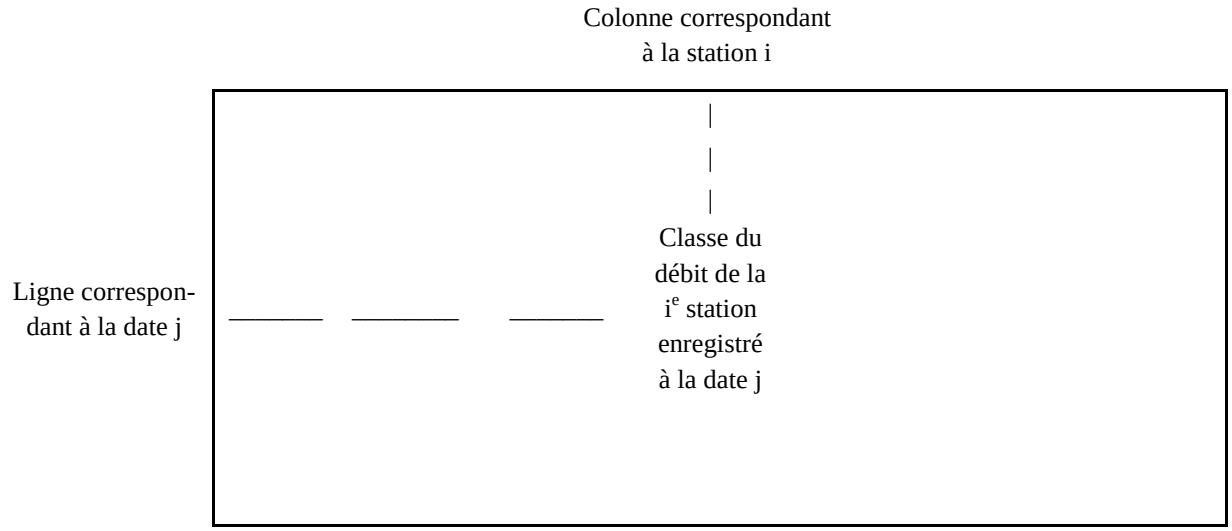


Tableau 10 : Présentation de la matrice M_histo

Etape 2 : Calcul du nombre d'occurrences de chaque événement observé

Une fois les données de débit transformées en classes correspondantes, on peut construire la loi conjointe.

Soit n le nombre de stations considérées.

Les événements observés et leur probabilité sont stockés dans une matrice appelée *proba*.

Elle se présente sous la forme :

Un événement est un vecteur ligne de taille n et correspond aux n premières colonnes de chaque ligne de la matrice

1er événement observé	probabilité
2e événement observé	probabilité
….	…
….	…
Dernier événement observé	probabilité

Tableau 11 : Présentation de la matrice proba

Le nombre de colonnes de cette matrice est égal à $(n+1)$. Sur les n premières colonnes de chaque ligne se trouvent les classes des n stations, ce qui correspond à un événement. La dernière colonne présente la probabilité de cet événement.

Le nombre de lignes de cette matrice est égal au nombre d'événements observés et s'incrémente au fur et à mesure de l'algorithme. Si on n'a observé qu'un événement, cette matrice a donc une seule ligne.

L'algorithme de construction de cette matrice est :

```
Initialiser la matrice proba

Pour chaque ligne de M_histo

      On regarde l'événement E associé à la ligne

      Si E est déjà dans la matrice proba, augmenter d'une unité la dernière co-
      lonne de la ligne correspondant à E (dans la matrice proba)

      Si E n'est pas dans la matrice proba, ajouter une ligne à la matrice proba
      correspondant à l'événement E et initialiser la dernière colonne de cette
      nouvelle ligne à 1.

Diviser le nombre d'occurrences, c'est à dire la dernière colonne de proba, par
le nombre de dates d'observation.
```

Nous donnons ci-dessous l'implémentation Matlab de cet algorithme :

```
proba=M_histo(1,1:nb_stations) ;
proba(1,nb_stations+1)=1 ;

nb_proba=1;

for d=2 :nb_dates

    existe=0;
    i=0;

    for i=1:nb_proba

        if M_histo(d,1:nb_stations)-proba(i,1:nb_stations)==zeros(1,nb_stations)

            proba(i,nb_stations+1)=proba(i,nb_stations+1)+1;
            existe=1;
            break
        end
    end

    if existe==0

        nb_proba=nb_proba+1;
        proba(nb_proba,1:nb_stations+1)=zeros(1,nb_stations+1);
        proba(nb_proba,1:nb_stations)=M_histo(d,1:nb_stations);
        proba(nb_proba,nb_stations+1)=1;
    end
end
```

Le résultat de ce programme est la matrice *proba*. C'est cette matrice que nous exploitons par la suite.

d. Temps de calcul de l'exécution du programme

Nous testons le programme du calcul de la loi conjointe sur un ordinateur ayant les performances suivantes :

- Fréquence du processeur : 2 GHz
- Mémoire vive : 4 Go

Nous avons fabriqué des données factices de débits par des tirages aléatoires entre 0 et 1, pour 300 stations et 15 000 dates. Nous avons pris pour toutes ces stations les 10 classes suivantes :

[0 , 0.1[	[0.1 , 0.2[	[0.2 , 0.3[	[0.3 , 0.4[	[0.4 , 0.5[
[0.5 , 0.6[	[0.6 , 0.7[	[0.7 , 0.8[	[0.8 , 0.9[	[0.9, 1]

Nous ne nous intéressons seulement au temps de calcul de la loi, sans tenir compte de l'importation des données depuis Excel et de l'affichage. Lorsque nous faisons varier le nombre de stations, nous obtenons les résultats suivants :

Nombre de stations prises en compte	Temps de calcul en secondes
2	2
3	23
4	93
5	150
6	190
10	308
20	481
50	737
100	1131
200	2074
300	2866

Tableau 12 : Evolution du temps de calcul en fonction du nombre de stations considérées pour la construction de la loi conjointe

On remarque que, pour cette quantité de dates (15 000 jours, soit environ 41 années d'historique), les temps de calcul sont tout à fait acceptables. L'exécution de la construction de la loi conjointe de 300 stations prend moins d'une heure.

Remarque sur la précision nécessaire

Dans notre exemple d'application, nous disposons d'au plus 100 ans de mesure, ce qui correspond à 36 500 dates. Dans le cas où l'on observe à chaque date un événement différent, chaque événement observé aura pour probabilité $\frac{1}{36\,500} = 2.7 \times 10^{-5}$.

Matlab est capable de gérer la précision requise par un tel ordre de grandeur.

B. Exploitation de la loi conjointe

L'objectif de la CCR est de pouvoir obtenir à partir de la loi conjointe calculée :

- La probabilité d'un événement choisi ;
- Le tirage aléatoire d'un événement selon la loi conjointe.

a. Obtenir la probabilité d'un événement ou d'un n-uplet de débit

On obtient la probabilité d'un événement par lecture dans la matrice *proba*. Le programme Matlab est la fonction `p_evenement.m` :

```
function y=p_evenement(x,A);

bornes=xlsread('donnees_reconstituees.xls','Feuil2');

taille=size(A);

nb_stations=taille(2)-1;

dim_bornes=size(bornes);

if nb_stations~=length(x)
    error('Veuillez vérifier la taille de l''évènement à tester');
end

isnull=1;

if length(x)==size(A,2)-1
    for i=1:size(A,1)
        if A(i,1:end-1)==x
            y=A(i,end);
            isnull=0;
        end
    end
    if isnull==1
        y=0;
    end
else y=0;
end

if y==0
    display('L''évènement testé n''a jamais été observé')
else proba_evenement=y
end
```

Cette fonction prend en entrée un vecteur x de taille n et une matrice A dont le nombre de colonnes est $(n+1)$.

On commence par enregistrer la matrice des bornes des intervalles contenue dans un fichier Excel, puis on compte le nombre d'intervalles par bornes. On vérifie ensuite si l'évènement à tester est de la bonne taille ; c'est-à-dire que le nombre de stations considéré pour l'évènement dont on veut connaitre la probabilité est bien n.

Dans l'étape suivante, on s'assure que les classes saisies pour l'évènement à tester existent dans le tableau des intervalles. On recherche dans les lignes de A si le vecteur x apparait aux n premières colonnes. Si c'est le cas, la fonction renvoie la valeur de la dernière colonne de la ligne. Sinon, la fonction renvoie 0.

Exemple :

$$x = (1,2,2)$$

$$y = (2,1,9)$$

$$A = \begin{pmatrix} 1 & 2 & 1 & 0,5 \\ 1 & 1 & 1 & 0,3 \\ 1 & 2 & 2 & 0,2 \end{pmatrix}$$

On aura $p\_evenement(x, A) = 0,2$ et $p\_evenement(y, A) = 0$.

Remarque : si la fonction renvoie 0, c'est que l'évènement n'a jamais été observé ; un message s'affiche pour l'indiquer.

b. *Effectuer un tirage aléatoire d'un événement à partir de la loi conjointe*

Illustrons d'abord la méthode utilisée par un exemple.

Supposons que l'exécution du programme de calcul de la loi conjointe ait donné la matrice :

$$proba = \begin{pmatrix} 1 & 2 & 1 & 0,5 \\ 1 & 1 & 1 & 0,3 \\ 1 & 2 & 2 & 0,2 \end{pmatrix}$$

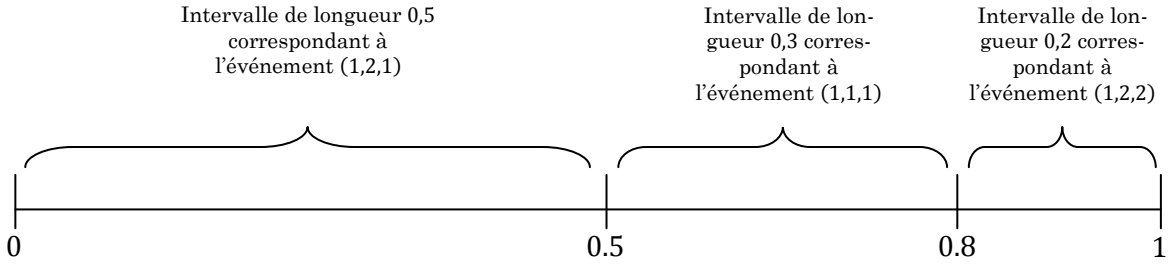
Ceci correspond à une loi conjointe sur 3 stations de mesures.

La lecture ligne par ligne de la matrice nous montre que les événements observés sont $(1,2,1), (1,1,1), (1,2,2)$ ayant pour probabilités respectives 0,5 , 0,3 et 0,2.

Par construction, la somme des probabilités des événements observés fait toujours 1.

Nous découpons l'intervalle [0,1] en 3 intervalles correspondants aux 3 événements observés, avec la condition que la longueur de chaque intervalle soit égale à la probabilité de l'événement qui lui correspond.

Dans notre exemple, cela donne : $[0\,;1] = [0\,;0{,}5] \cup [0{,}5\,;\,0{,}8] \cup [0{,}8\,;\,1]$.



Ce découpage de l'intervalle [0,1] se fait en cumulant les probabilités de la dernière colonne de la matrice *proba*. On stocke ces sommes cumulées dans la matrice *repart*.

Dans notre exemple cela donne : $repart = \begin{pmatrix} 1 & 2 & 1 & 0{,}5 \\ 1 & 1 & 1 & 0{,}8 \\ 1 & 2 & 2 & 1 \end{pmatrix}$

Si nous voulons maintenant obtenir un événement selon la loi conjointe, on choisi « au hasard » une valeur t entre 0 et 1. Se présentent alors 3 possibilités :

- $t \in [0\,,0{,}5]$: c'est l'événement (1,2,1) qui est obtenu ;
- $t \in [0{,}5\,,0{,}8]$: c'est l'événement (1,1,1) qui est obtenu ;
- $t \in [0{,}8\,,1]$: c'est l'événement (1,2,2) qui est obtenu.

Nous disposons sous Matlab de la fonction `rand()` qui donne un nombre aléatoire entre 0 et 1 de manière uniforme.

Ainsi, la probabilité qu'un appel de la fonction `rand()` tombe dans l'un des 3 intervalles est égale à la longueur de cet intervalle.

En pratique, l'algorithme est :

```
Construire la matrice repart des probabilités cumulées

Tirer aléatoirement une valeur t entre 0 et 1.

Déterminer entre quelles valeurs de la dernière dans lequel se trouve t
```

Le programme Matlab associé est :

```
repart=proba;

repart(:,end)=cumsum(proba(:,end));

t=rand();

indice=num_inter(t,repart(:,end))+1;

evenement=proba(indice,1:end-1)
```

Remarquons que nous utilisons à nouveau la fonction `num_inter.m` pour déterminer l'intervalle dans lequel se trouve *t*.

c. *Tirage d'un n-uplet de débit*

La CCR souhaite simuler des *n*-uplets de débits, et non des *n*-uplets de classes comme le fait le programme du paragraphe précédent (III. B. b).

Cependant, l'hypothèse émise lors du découpage des plages de valeurs de débits en différentes classes consistait à dire que toutes les valeurs au sein d'une même classe avaient la même probabilité d'apparition. C'est pourquoi obtenir une valeur de débit à partir d'un intervalle revient effectuer un tirage uniforme entre les bornes de celui-ci.

Le principe est donc de tirer aléatoirement un événement comme présenté dans la partie III.B.b, puis pour chaque composante de l'événement, tirer aléatoirement de manière uniforme une valeur dans la classe correspondante.

Le programme réalisant cette opération est :

```
bornes=zeros(2,nb_stations);

for l=1:nb_stations

    bornes(:,l)=interv(evenement(l)+1:evenement(l)+2,l) ;

end

nuplet=zeros(1,nb_stations);

for i=1:nb_stations

    x=rand();
    nuplet(i)=(1-x) * bornes(1,i) + bornes(2,i)*x;

end
```

La première boucle `for` construit à partir des données des intervalles, une matrice à 2 lignes et $(n+1)$ colonnes (rappelons que n est le nombre de stations).

La première ligne de la matrice contient les bornes inférieures des intervalles, la deuxième ligne les bornes supérieures. La seconde boucle `for` tire aléatoirement de manière uniforme une valeur entre les bornes construites par la première boucle.

C. Mise en œuvre sur un exemple simple

Considérons une étude portant sur 10 jours et 3 stations.

Les valeurs de débits sont données par la matrice :

$$donnees = \begin{pmatrix} & 1 & 2 & 3 \\ 1 & 2{,}5 & 3{,}2 & 1{,}3 \\ 2 & 2{,}5 & 1{,}8 & 2{,}2 \\ 3 & 3{,}1 & 3{,}3 & 1{,}1 \\ 4 & 4{,}2 & 1{,}7 & 1{,}7 \\ 5 & 4{,}8 & 3{,}2 & 5{,}1 \\ 6 & 2{,}8 & 2{,}1 & 1{,}5 \\ 7 & 4{,}3 & 2{,}9 & 2{,}9 \\ 8 & 4 & 3{,}3 & 3{,}5 \\ 9 & 4{,}5 & 3{,}9 & 4{,}8 \\ 10 & 0{,}2 & 0{,}8 & 0{,}8 \end{pmatrix}$$

Nous prenons 2 classes pour chaque station, données par la matrice :

$$interv = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 0 & 0 \\ 5 & 2 & 4 \\ 10 & 4 & 6 \end{pmatrix}$$

Par exemple pour la station 1, les deux classes considérées sont [0,5[et [5,10].

La transformation des données en classes donne :

$$M\_histo = \begin{pmatrix} & 1 & 2 & 3 \\ 1 & 1 & 2 & 1 \\ 2 & 1 & 1 & 1 \\ 3 & 1 & 2 & 1 \\ 4 & 1 & 1 & 1 \\ 5 & 1 & 2 & 2 \\ 6 & 1 & 2 & 1 \\ 7 & 1 & 2 & 1 \\ 8 & 1 & 2 & 1 \\ 9 & 1 & 2 & 2 \\ 10 & 1 & 1 & 1 \end{pmatrix}$$

Nous avons $2^3 = 8$ événements observables qui sont :

(1,1,1)	(2,1,1)
(1,1,2)	(2,1,2)
(1,2,1)	(2,2,1)
(1,2,2)	(2,2,2)

Les événements effectivement observés sont :

- (1,2,1) : observé aux dates $1, 3, 6, 7, 8$, soit 5 fois parmi les 10 dates de l'historique ;
- (1,1,1) : observé aux dates $2, 4, 10$, soit 3 fois parmi les 10 dates de l'historique ;
- (1,2,2) : observé aux dates $5, 9$, soit 2 fois parmi les 10 dates de l'historique.

Nous obtenons la matrice :

$$proba = \begin{pmatrix} 1 & 2 & 1 & 0{,}5 \\ 1 & 1 & 1 & 0{,}3 \\ 1 & 2 & 2 & 0{,}2 \end{pmatrix}$$

A partir de cette matrice, et donc de la loi conjointe calculée, nous voulons obtenir un triplet de débits.

Pour cela, nous utilisons la matrice des probabilités cumulées :

$$repart = \begin{pmatrix} 1 & 2 & 1 & 0{,}5 \\ 1 & 1 & 1 & 0{,}8 \\ 1 & 2 & 2 & 1 \end{pmatrix}$$

Nous tirons un nombre aléatoirement entre 0 et 1 suivant la loi uniforme.

$$t = 0{,}567$$

Comme nous avons $0{,}5 < t < 0{,}8$, l'événement obtenu est (1,1,1).

Nous construisons la matrice correspondant aux bornes des intervalles correspondant aux classes $1, 1$ et 1.

$$bornes = \begin{pmatrix} 0 & 0 & 0 \\ 5 & 2 & 4 \end{pmatrix}$$

Cette matrice se lit :

- La classe 1 de la station 1 correspond à l'intervalle [0 , 5] ;
- La classe 1 de la station 2 correspond à l'intervalle [0 , 2] ;
- La classe 1 de la station 3 correspond à l'intervalle [0 , 4].

A présent nous effectuons un tirage uniforme sur chacun des intervalles [0 , 5], [0 ,2], [0 , 4].

Nous obtenons le triplet : (1,3 , 0,2 , 3,2).

D. Utilisation des probabilités conditionnelles

La CCR souhaite pouvoir calculer la probabilité d'évènements en utilisant les probabilités conditionnelles : on suppose que l'on connaît la probabilité d'observer une certaine classe sur une station donnée. Si la loi conjointe ne donne pas le même résultat, l'objectif est de rectifier les probabilités des évènements tels que le débit de la station appartient à la classe étudiée.

Nous avons créé une fonction prenant en entrée un nom de station A, une classe et la probabilité que le débit de A appartienne à cette classe. La fonction extrait de la loi conjointe tous les évènements pour lesquels le débit de A appartient à cette classe et recalcule la probabilité de ces évènements en utilisant les probabilités conditionnelles.

$$P(X_A \cap X_1, \dots, X_{n-1}) = P(X_A) \times P(X_1, \dots, X_{n-1} | X_A)$$

Où :

- $P(X_A)$ est la probabilité que X_A se produise, c'est-à-dire d'observer une certaine classe de débit sur la station A ;

- $P(X_1, \dots, X_{n-1} | X_A)$ est la probabilité d'observer $X_1, \dots, X_{n-1}$ sachant que X_A se produit. Cette probabilité est calculée lors de la construction de la loi conjointe ;

- $P(X_A \cap X_1, \dots, X_{n-1})$ est la probabilité d'observer simultanément les évènements X_A et $X_1, \dots, X_{n-1}$. C'est la probabilité que nous cherchons à connaître.

Exemple :

Nous disposons du tableau de classes suivant :

$$M\_histo = \begin{pmatrix} & 1 & 2 & 3 \\ 1 & 1 & 2 & 1 \\ 2 & 1 & 1 & 1 \\ 3 & 1 & 2 & 1 \\ 4 & 1 & 1 & 1 \\ 5 & 1 & 2 & 2 \\ 6 & 1 & 2 & 1 \\ 7 & 1 & 2 & 1 \\ 8 & 1 & 2 & 1 \\ 9 & 1 & 2 & 2 \\ 10 & 1 & 1 & 1 \end{pmatrix}$$

Nous savons que la probabilité que le débit de la station 2 appartienne à la classe 2 est de 0.1. Or le calcul de la loi conjointe nous indique une probabilité différente. L'objectif est de recalcu-

ler la probabilité des évènements tels que le débit de la station 2 appartienne à la classe 2, de telle sorte que la probabilité globale de cette classe vaille 0.1.

Nous réduisons donc notre tableau aux dates où la classe 2 a été observée sur la station 2.

$$M\_histo = \begin{pmatrix} & 1 & 2 & 3 \\ 1 & 1 & 2 & 1 \\ 3 & 1 & 2 & 1 \\ 5 & 1 & 2 & 2 \\ 6 & 1 & 2 & 1 \\ 7 & 1 & 2 & 1 \\ 8 & 1 & 2 & 1 \\ 9 & 1 & 2 & 2 \end{pmatrix}$$

Les évènements observés sont : (1,2,1) et (1,2,2).

Le calcul de la loi conjointe donne les résultats suivants :

- La probabilité d'observer l'évènement (1,2,1) est 0,714 ;
- La probabilité d'observer l'évènement (1,2,2) est 0,286.

Pour obtenir les probabilité de (1,2,1) et de (1,2,2) sur tout l'historique, il ne nous reste qu'à multiplier les probabilités obtenues par la probabilité que le débit mesuré sur la station 2 appartienne à la classe 2, c'est-à-dire 0,1.

La probabilité de l'évènement (1,2,1) est donc de 0,0714 ; et celle de (1,2,2) est de 0,0286.

a. Algorithme mis en place

Pour calculer les probabilités d'évènements en utilisant cette méthode nous procédons de la façon suivante :

```
Saisir la référence de la station considérée, la classe étudiée et la probabilité de l'évènement.

Parcourir tout l'historique pour isoler les dates où l'évènement se produit.

Appliquer le calcul de la loi conjointe sur ce nouveau tableau.

Multiplier les probabilités obtenues pour ce tableau par celle de l'évènement considéré

Renvoyer les résultats sous forme d'un tableau
```

b. *Implémentation Matlab*

Le code Matlab effectuant ces actions reprend la structure du code calculant la loi conjointe, auquel nous ajoutons des étapes et plusieurs fonctions.

Le début du programme n'est pas modifié, jusqu'au remplacement dans le tableau de débits des valeurs par des numéros de classe. Nous avons ensuite ajouté les fonctions suivantes :

- **`evènement.m`** : fonction demandant à l'utilisateur de saisir la station étudiée, la classe de débit, et la probabilité que le débit de la station appartienne à cette classe. La fonction vérifie également si les données saisies sont cohérentes ;

- **`extraction.m`** : fonction permettant d'extraire les valeurs de classe de l'historique pour la station considérée et les bornes d'intervalles qui lui sont associées ;

- **`selection_dates.m`** : fonction repérant et stockant les dates où la classe saisie par l'utilisateur a été observée pour la station considérée ;

- **`reduction_dates.m`** : fonction créant un tableau dans lequel seules les dates où l'évènement considéré est observé apparaissent ;

Nous détaillons à présent les codes Matlab de chacune de ces fonctions, ainsi que leurs données d'entrée et de sortie.

- **`evènement.m`**

Cette fonction permet à l'utilisateur de saisir le nom de la station considérée, la classe étudiée, et la probabilité d'observer cette classe sur cette station. Elle vérifie également que les données saisies sont cohérentes.

En entrée, la fonction reçoit un vecteur stockant les noms de toutes les stations et un tableau gardant en mémoire pour chaque station le nombre d'intervalles. Elle renvoie un vecteur appelé « évènement » et qui stocke le nom de la station saisi, son indice dans le tableau de données, la classe entrée par l'utilisateur et sa probabilité.

```
function new_probas=evenement(stations, bornes)

nb_colonnes=size(bornes,2);
station=NaN;
classe=NaN;
proba_evenement=NaN;

while isnan(station)==1
    station=input('Station à isoler?');
end
```

```
cherche_colonne=0;
for i=1:nb_colonnes
    if station==stations(i)
        cherche_colonne=i;
    end
end

while cherche_colonne==0
    display('La station saisie ne fait pas partie des stations de la table des
débits');
    station=input('Station à isoler?');
    for i=1:nb_colonnes
        if station==stations(i)
            cherche_colonne=i;
        end
    end
end

while isnan(classe)==1
    classe=input('Classe de la station?');
end

cherche_borne=0;

for i=1:bornes(2,i)-1
    if classe==i
        cherche_borne=1;
    end
end

while cherche_borne==0
    display('Le numéro de classe saisi est incorrect, il doit être strictement
supérieur à 0 et au  maximum égal à:');
    bornes(2,i)-1
    classe=input('Classe de la station?');
    for i=1:bornes(2,i)-1
        if classe==i
            cherche_borne=1;
        end
    end
end

while (isnan(proba_evenement)==1)
    proba_evenement=input('Probabilité de l''évènement traité?');
end

verif_proba=0;

if (proba_evenement<=1) & (proba_evenement>0)
    verif_proba=1;
end

while verif_proba==0
```

```
    display('La probabilité doit être strictement supérieure à 0 et au maximum
égale à 1');
    proba_evenement=input('Probabilité de l''évènement traité?');
    if (proba_evenement<=1) & (proba_evenement>0)
        verif_proba=1;
    end
end

while proba_evenement>1
    display('Une probabilité ne peut pas être supérieure à 1');
    proba_evenement=input('Probabilité de l''évènement traité?');
end

while proba_evenement<=0
    display('La probabilité doit être supérieure à 0');
    proba_evenement=input('Probabilité de l''évènement traité?');
end

new_probas=[station cherche_colonne classe proba_evenement];
```

- **extraction.m**

La fonction reçoit en entrée : un vecteur événement, le tableau donnant les classes des débits et le vecteur stockant les noms des stations.

Le résultat renvoyé est un vecteur stockant les classes de débit recensées sur la station saisie par l'utilisateur.

```
function donnees_station=extraction(evenement,tab_classes,stations)

nb_colonnes=length(stations);

station_choisie=evenement(1);

indice_station=0;

for i=1:nb_colonnes
    if station_choisie==stations(i)
        indice_station=i;
    end
end

donnees_station=tab_classes(:,indice_station);
```

- **selection_dates.m**

Cette fonction, qui répertorie les lignes où la classe saisie par l'utilisateur, reçoit en entrée un vecteur appelé « évènement », le vecteur stockant toutes les classes observées pour la station considérée.

Le résultat obtenu se présente sous la forme d'un vecteur où sont stockés les indices des lignes où la classe saisie par l'utilisateur a été observée.

```
function selection=selection_dates(evenement, station);

nb_dates=length(station);
classe_evenement=evenement(3);
nb_occurrences=0;

for i=1:nb_dates
    if station(i)==classe_evenement
        nb_occurrences=nb_occurrences+1;
        selection(nb_occurrences)=i;
    end
end

if nb_occurrences==0
    error('La classe renseignée pour cet évènement n''a jamais été observée');
end
```

- **reduction_dates.m**

Le but de cette fonction est de réduire le tableau de données contenant tout l'historique de mesures uniquement aux dates où la classe considérée a été observée. Elle reçoit comme données d'entrée le vecteur stockant les indices correspondant à ces dates, et le tableau contenant les valeurs de classes pour toutes les stations sur tout l'historique.
Le résultat obtenu se présente sous la forme d'un tableau de classes réduit aux seules dates où la classe saisie par l'utilisateur a été observée sur la station considérée.

```
function just_dates=reduction_dates(dates_eve, tab_donnees)

nb_dates_eve=length(dates_eve);

for i=1:nb_dates_eve
    just_dates(i,:)=tab_donnees(dates_eve(i),:);
end
```

E. Fonctions Matlab supplémentaires

A la demande de la CCR, différentes fonctions Matlab supplémentaires ont été créées, permettant la mise en place de la construction de la loi conjointe.

a. Recherche d'un évènement

Nous avons créé un fonction qui demande à l'utilisateur d'entrer un évènement sous forme de classes, puis renvoie les différentes dates auxquelles il a été observé ainsi que les débits observés à ces dates.

L'algorithme mis en place est le suivant :

```
Comparer chaque ligne du tableau de classes avec l'évènement testé

Si la ligne est identique à l'évènement :

      Stocker dans le tableau de résultats la date et les débits mesurés à cette
      date

      Passer à la ligne suivante

Si l'évènement n'a jamais été observé, le signaler à l'utilisateur

Une fois le tableau entièrement parcouru, renvoyer les résultats dans un tableau
Excel.
```

Le programme qui a été créé renvoie un tableau se présentant sous la forme suivante : la première colonne correspond aux dates où l'évènement testé a été observé, les colonnes suivantes donnent les débits relevés à chaque station à ces dates.

Il est obligatoire d'utiliser ce programme après avoir effectué la construction de la loi conjointe, ou après avoir utilisé le programme utilisant les probabilités conditionnelles. En effet, plusieurs des données nécessaires à la recherche d'un évènement sont créées par ces programmes : par exemple, la matrice où les données de débit ont été transformées en classes.

Le code Matlab est le suivant :

```
evenement= input('Rentrer l''evenement a rechercher sous forme de classes')

if nb_stations~=length(evenement)
    error('Veuillez vérifier la taille de l''évènement à rechercher');
end

nb_occurence=0;

for i=1:nb_dates
```

```
    if evenement(1,1:nb_stations)==M_histo_fixe(i,1:nb_stations)
        nb_occurence=nb_occurence+1;
        re-
sults(nb_occurence,1:nb_stations+1)=[dates(i),donnees(i,1:nb_stations)];
    end

end

if nb_occurence==0
    display('L''evenement n''a jamais été observé');
end

if nb_occurence~=0
    display('consulter le fichier result_recherche.xls pour accéder aux résul-
tats')
    xlswrite('result_recherche.xls',results);
end
```

b. Programme permettant le calcul de probabilités sur une station unique

A la demande de la CCR, nous avons créé un programme permettant de calculer la probabilité d'une classe donnée pour une station donnée. Il s'agit donc simplement d'isoler le vecteur des classes pour une station saisie par l'utilisateur, et de calculer la probabilité de l'évènement considéré.

Le code implémenté sous Matlab à cet effet est le suivant :

```
clear all

donnee=xlsread('donnees_reconstituees.xls','Feuil1');

taille=size(donnee);
nb_dates=taille(1)-1;
nb_stations=taille(2)-1;
stations=donnee(1,2:taille(2));
dates=donnee(2:taille(1),1);

interv=xlsread('donnees_reconstituees.xls','Feuil2');

clear nb_bornes

for i=1:nb_stations
    nb_bornes(1,i)=stations(i);
    nb_bornes(2,i)=0;
    for j=2:size(interv,1)
        if isnan(interv(j,i))==0
            nb_bornes(2,i)=nb_bornes(2,i)+1;
        end
```

```
    end
end

nb_bornes;

w=verif_stations(donnee,interv,nb_stations);

if w~=0
    error('Merci de vérifier les références des stations a étudier et l ordre
dans lequel elles sont ordonnées dans le document Excel puis de relancer le pro-
gramme')
end

dim(1)=max(nb_bornes(2,1:nb_stations));

donnees=donnee(2:nb_dates+1,2:nb_stations+1);
interv=interv(2:dim(1)+1,1:nb_stations);

vi=verif_bornes(interv,nb_bornes,nb_stations);

for i=1:nb_stations
    if vi(i)~=0
        mauvaise_station=stations(i)
        error('Merci d''ordonner les bornes de intervalles renseignées pour la
station ci-dessus dans le document Excel en ordre croissant')
    end
end

vb=verif_debit(donnees,interv,nb_stations,nb_bornes);

for i=1:nb_stations
    if vb(i)~=0
        mauvaise_station=stations(i)
        error('Merci de vérifier les valeurs de débits rentrées ou de changer
les bornes des intervalles de la station mentionnée ci-dessus')
    end
end

station=NaN;
classe=NaN;

while isnan(station)==1
    station=input('Station à traiter?');
end

cherche_colonne=0;
for i=1:nb_stations
    if station==stations(i)
        cherche_colonne=i;
    end
end

while cherche_colonne==0
```

```
    display('La station saisie ne fait pas partie des stations de la table des
débits');
    station=input('Station à isoler?');
    for i=1:nb_stations
        if station==stations(i)
            cherche_colonne=i;
        end
    end
end

while isnan(classe)==1
    classe=input('Classe de la station?');
end

cherche_borne=0;

for i=1:nb_bornes(2,i)-1
    if classe==i
        cherche_borne=1;
    end
end

while cherche_borne==0
    display('Le numéro de classe saisi est incorrect, il doit être strictement
supérieur à 0 et au  maximum égal à:');
    nb_bornes(2,i)-1
    classe=input('Classe de la station?');
    for i=1:nb_bornes(2,i)-1
        if classe==i
            cherche_borne=1;
        end
    end
end

evenement_A=[station cherche_colonne classe];

debits_station=donnees(:,cherche_colonne);

intervalles_station=interv(:,cherche_colonne);

count_suppr=0;

count_nan=0;

for i=1:nb_dates
    if isnan(debits_station(i))==1
        a_suppr(count_nan+1)=i;
        count_nan=count_nan+1;
    end
end

if count_nan~=0
    a_suppr=unique(a_suppr);
    nb_suppr=length(a_suppr);
```

```
    for i=1:nb_suppr
        donnee=suppr_lignes(debits_station,nb_dates,1,a_suppr(i)-(i-1));
        nb_dates=nb_dates-1;
    end
end

debits_station=debits_station(1:nb_dates,1);

M_histo(1:nb_dates,1)=num_inter(debits_station(1:nb_dates,1),intervalles_station
(1:nb_bornes(2,cherche_colonne),1));

proba(1,1)=evenement_A(3) ;
proba(1,2)=0 ;

nb_proba=0;
for d=1 :nb_dates
    if M_histo(d,1)-proba(1,1)==zeros(1,1)
            proba(1,2)=proba(1,2)+1;
            nb_proba=nb_proba+1;
    end
end
nb_proba;

% Normalisation probas

if nb_proba==0
    display('La classe saisie n''a jamais été observée pour la station trai-
tée');
else
    proba(1,2)=proba(1,2)/nb_dates;
    display('La probabilité recherchée est de :');
    proba(1,2)
end
```

IV. Reconstitution de données manquantes

Les données de débit dont nous disposons sont incomplètes : à certaines dates, la mesure n'a pas été effectuée, est incorrecte ou n'a pas été enregistrée dans la base de données.

Pour pallier cette absence de relevés, nous avons développé une méthode de reconstruction des données manquantes basée sur la théorie présentée dans le livre [2].

Cette étape doit intervenir avant la construction de la loi conjointe.

A. Méthode de reconstruction

Pour reconstruire les données sur une station, nous utilisons les données disponibles sur d'autres stations.

La base de données contient 333 stations mais il n'est pas pertinent de toutes les utiliser. On détermine, pour chaque station à reconstituer, la liste des stations à prendre en compte pour la reconstruction, c'est-à-dire la liste des stations les plus corrélées.

Une fois que nous connaissons cette liste, nous utilisons une méthode probabiliste robuste de reconstructions des données faisant intervenir les espérances conditionnelles : une fois l'intervalle optimal déterminé, on remplace chaque valeur manquante par l'espérance conditionnelle selon la valeur d'une station corrélée à la même date.

a. Calcul des corrélations et extraction des stations les mieux corrélées

La méthode utilisée classiquement pour déterminer la corrélation entre deux séries de données $X = x_1 \dots x_i \dots x_n$ et $Y = y_1 \dots y_i \dots y_n$ est le calcul du coefficient de corrélation linéaire $\rho(X,Y)$ selon la formule :

$$\rho(X,Y) = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2}\sqrt{\sum_i y_i^2}}$$

avec x_i et y_i centrés sur la moyenne ; c'est-à-dire : $x_i = x_i - \bar{x}$ et $y_i = y_i - \bar{y}$

Cette formule s'applique à des échantillons de même taille. Or, dans notre cas, du fait des données manquantes, nous ne disposons pas du même nombre de mesures de débits pour chaque rivière et les séries n'ont pas la même taille.

Nous sommes donc amenés à utiliser un autre coefficient de corrélation, non symétrique, et qui fait intervenir les domaines d'existence de chaque rivière (voir [2], p.100).

Ce coefficient est défini par la formule :

$$\rho_X(Y) = \frac{\sum_{i \in D(X) \cap D(Y)} x_i y_i}{\sqrt{\sum_{i \in D(X)} x_i^2} \sqrt{\sum_{i \in D(X) \cap D(Y)} y_i^2}}$$

où, de la même manière, x_i et y_i centrés sur la moyenne.

Pour reconstruire la station *X*, nous utilisons les stations *Y* pour lesquelles $\rho_X(Y)$ est le plus proche de 1.

Nous fixons un seuil de corrélation minimum acceptable, par exemple 0.8, et nous ne considérons que les stations *Y* telles que $\rho_X(Y)$ est supérieur à ce seuil. Nous cherchons donc les meilleures corrélations pour chaque fonction.

Les étapes décrites ensuite sont effectuées pour chaque couple de stations considérées comme bien corrélé, jusqu'à ce que toutes les valeurs manquantes soient reconstituées, ou bien qu'il ne reste plus de couple bien corrélé.

Pour chaque station, nous commençons par reconstituer les données manquantes à partir de la station la plus corrélée, puis, s'il reste des données non reconstituées, nous utilisons la deuxième, puis la troisième …

b. Calcul du NEVD optimal

La reconstruction de valeurs manquantes se fait à l'aide d'intervalles qui peuvent être définis suivant plusieurs méthodes. . En effet, ils peuvent être construits de différentes façons, en fixant l'un de ces paramètres :

- la longueur de l'intervalle ;
- le nombre de points contenu dans l'intervalle ;
- le nombre de valeurs distinctes dans l'intervalle.

Les intervalles obtenus ne seront pas les mêmes suivant la méthode utilisée pour les construire.

La méthode donnant la reconstruction la plus précise est l'intervalle construit par la méthode du nombre fixe de valeurs distinctes. Ce nombre sera noté NEVD, soit « Nombre Egal de Valeurs Distinctes ».

Pour la reconstruction, nous allons donc diviser le vecteur qui sert à reconstruire les données manquantes en intervalles, contenant tous le même nombre de valeurs. Pour un vecteur donné, il existe autant de possibilités de découpage qu'il possède de valeurs distinctes. Il s'agit donc de définir le NEVD optimal ; c'est-à-dire celui qui donnera la meilleure reconstruction.

Le NEVD ou Nombre Egal de Valeurs distinctes optimal est obtenu à l'aide d'une série d'opérations. On considère que le vecteur X est à reconstruire et Y est le vecteur utilisé dans ce but.

Les différentes actions effectuées pour déterminer le NEVD optimal sont les suivantes :

a. Extraction des données communes entre X et Y
b. Séparation des vecteurs en 2 périodes
c. Calcul du Nombre de Valeurs Distinctes dans la 1ère période de Y (qu'on notera Y_{C1})

Les opérations suivantes sont effectuées pour chaque NEVD possible pour Y_{C1} ; c'est-à-dire de 1 au nombre calculé en c.

d. Séparation de Y_{C1} en intervalles associés au nombre de valeurs distinctes
e. Construction de la table des espérances conditionnelles de X_{C1} en fonction de Y_{C1}
f. Reconstruction de X_{C2} à l'aide de la table des espérances conditionnelles
g. Calcul de l'indicateur de proximité

Une fois ces opérations effectuées, on compare les indicateurs de proximité : le NEVD optimal est défini par l'indicateur de proximité le plus faible. C'est cette valeur de NEVD qu'on choisit pour la reconstruction de X par Y.

Plus précisément, les différentes étapes sont les suivantes :

a. Extraire les données communes de X et Y consiste à isoler les intervalles sur lesquels les vecteurs X et Y existent tous les deux, c'est-à-dire $D(X \cap Y)$ et à extraire les valeurs de X et Y appartenant à ces intervalles pour les stocker dans deux nouveaux vecteurs notés X_C et Y_C.

b. Nous séparons ensuite chacun des vecteurs en deux périodes notées : X_{C_1}, X_{C_2}, Y_{C_1} et Y_{C_2}.

c. Nous considérons maintenant la première période de Y_C, que nous notons Y_{C_1}. Nous cherchons à en connaitre le nombre de valeurs distinctes.

Pour cela, nous devons éliminer les doublons du vecteur, et recenser le nombre de valeurs restant une fois cette opération effectuée.

Nous notons *max_nvd* le nombre de valeurs distinctes de Y_{C_1}.

Il s'agit ensuite d'obtenir le NEVD optimal.

Les étapes suivantes sont réalisées pour chaque valeur de NEVD possible, c'est-à-dire de 1 à *max_nvd* inclus. Nous notons *k_nvd* la valeur de NEVD considérée ($k_{nvd} = 1 \ldots max\_nvd$).

d. Nous divisons Y_{C1} en intervalles de telle sorte que chaque intervalle contienne *k_nvd*.valeurs distinctes.

Le but est d'obtenir un vecteur contenant les bornes des intervalles ainsi construits.

Tout d'abord nous calculons le Nombre d'EXtrémités d'intervalles (NEX). Le livre [2] nous donne en page 130 une formule pour le calcul du NEX que nous appliquons :

$$NEX = Partie\_entière\left(\frac{nvd\left(Y_{C1}\right)-1}{k\_nvd}\right)+1$$

Ce calcul nous donne la longueur du vecteur recherché. Pour construire ce vecteur en question, nous utilisons le vecteur des valeurs distinctes de Y_{C1}. Nous prenons la valeur de la première composante, puis la valeur des composantes toutes les *k_nvd* composantes. Nous obtenons ainsi un vecteur de NEX composantes, qui sont les extrémités des intervalles associées à Y_{C1} et à *k_nvd*.

Prenons par exemple :

$$Y_{C1} = (2{,}9{,}7{,}6{,}9{,}9{,}1, NaN, 1{,}7)$$

Nous cherchons les bornes des intervalles pour un nombre de valeurs distinctes égal à $k_{nvd} = 3$.

Nous obtenons :

$$nvd(Y_{C1}) = 5$$

$$NEX = Partie.entière\left(\frac{(5-1)}{3}\right) + 1 = 2$$

$$Y_{C1}\_distinct = (1{,}2{,}6{,}7{,}9)$$

Nous prenons les composantes de $Y_{C1}\_distinct$ toutes les 3 composantes :

$$vecteur\_resultat = (1{,}7)$$

e. Nous disposons maintenant des intervalles qui vont nous servir à calculer les espérances conditionnelles de X_{C1} en fonction de Y_{C1}.

La table des espérances conditionnelles d'un vecteur X en fonction de Y donne la valeur moyenne attendue pour X sachant Y, ou plus précisément l'intervalle auquel appartient Y. Elle se présente de la manière suivante :

Intervalle de Y	Espérance de X
$\left[a(1), a(2)\right[$	$b(1)$
$\left[a(2), a(3)\right[$	$b(2)$
$\left[a(3), a(4)\right[$	$b(3)$
Etc.	Etc.

Afin de calculer l'espérance de X sur un intervalle, on procède de la manière suivante : on recense tous les valeurs de Y appartenant à cet intervalle, et on calcule la moyenne des valeurs de X qui leur sont associés.

f. Nous avons désormais à notre disposition la table des espérances conditionnelles de X_{C1} en fonction de Y_{C1}. Nous procédons à la reconstruction de X_{C2} à l'aide de cette table.

Nous avons également besoin du vecteur Y_{C2} qui est parfaitement défini. Nous cherchons dans la table des espérances conditionnelles l'intervalle dans lequel est située chaque composante de Y_{C2}, ce qui nous permet d'obtenir l'espérance de X, et de remplacer chacune des composante de X_{C2} par la valeur donnée par la table des espérances conditionnelles.

g. Une fois le vecteur X_{C2} reconstruit, il est possible de calculer l'indice de proximité associé à cette reconstruction : il caractérise la qualité de la reconstruction. Il se calcule de la manière suivante :

$$IQ = \left(\frac{\sum_i \left(x_i - x_i^* \right)^2}{\sum_i x_i^2} \right)^{1/2}$$

où x_i^* est la valeur reconstruite, et i parcourt uniquement les intervalles où X a été reconstruit.

Plus l'indice est faible, meilleure est la reconstruction.

Les étapes a à g sont répétées pour chaque NEVD possible. Le NEVD optimal correspond à l'indice de proximité le plus faible.

c. Construction des intervalles de Y

L'étape précédente a permis de déterminer le NEVD optimal. Nous pouvons à présent procéder à la reconstruction du vecteur *X* à l'aide du vecteur *Y*.

On commence par reconstruire les intervalles de *Y*. Nous avons ainsi à notre disposition le vecteur donnant les bornes des intervalles qui vont être utilisés pour construire la table des espérances conditionnelles.

d. Calcul de l'espérance conditionnelle de X en fonction de Y

La construction de la table des espérances conditionnelles s'effectue à l'aide de la méthode définie dans le paragraphe *b* : pour chaque intervalle de *Y*, nous recensons les valeurs du vecteur qui en font partie, et nous calculons la moyenne des valeurs de *X* qui leur sont associées

Nous avons ainsi à notre disposition la table des espérances conditionnelles de *X* en fonction de *Y*, qui va nous permettre de reconstruire les valeurs manquantes de *X*.

e. Reconstruction des données manquantes

La dernière étape est la reconstruction des données manquantes. Pour chaque valeur manquante de *X*, nous regardons la valeur de *Y* lui correspondant, et nous allons chercher dans la table des espérances conditionnelles la valeur de *X* correspondant à l'intervalle où se trouve *Y*.

Si la valeur de Y ne correspond à aucun des intervalles de la table des espérances conditionnelles dont nous disposons, nous laissons la valeur de *X* non renseignée, et nous tenterons de la reconstruire à l'aide d'une autre station considérée comme bien corrélée.

Toutes ces étapes de reconstruction sont répétées jusqu'à ce que la station soit entièrement reconstruite, ou qu'il ne reste plus aucune station avec laquelle elle est considérée comme bien corrélée.

B. Algorithmes de reconstruction de données manquantes

a. Algorithme général

Pour la reconstruction des données manquantes, un algorithme reprenant les grands points de la méthodologie a été créé :

Calculer les corrélations entre les différentes stations.

Extraire les meilleures corrélations.

Pour chaque station :

    Reconstruire un maximum de données à l'aide des meilleures corrélations.

    Si toutes les stations bien corrélées ont été utilisées, ou si toutes les données ont été reconstruites, passer à la reconstruction de la station suivante.

Stocker et renvoyer les résultats.

b. Extraction des stations les mieux corrélées

Après calcul des coefficients à l'aide de la formule donnée dans la partie précédente, nous cherchons à isoler les meilleures corrélations pour chaque station. Nous fixons un seuil minimal pour le coefficient de corrélation : deux stations sont définies comme bien corrélées, si leur coefficient de corrélation est supérieur au seuil.

L'algorithme mis en place pour effectuer cette opération est le suivant :

Pour chaque station, parcourir tous les coefficients de corrélation qui lui sont associés,

    Si le coefficient est supérieur au seuil fixé le stocker,

    Passer à la station suivante,

    Une fois tous les coefficients parcourus trier ceux qui sont stockés en ordre décroissant,

    Remplacer les coefficients par le nom de la station qui leur est associée,

Renvoyer un tableau donnant les bonnes corrélations pour chaque station sur une ligne, de la meilleure à la moins bonne.

Les résultats sont stockés comme suit :

	Station bien corrélée 1	Station bien corrélée 2	Station bien corrélée 3	Station bien corrélée 4
Station 1	2	3	5	0
Station 2	1	3	5	4
Station 3	5	1	2	4
Station 4	5	3	2	1
Station 5	3	4	2	1

Ce tableau se lit par ligne : chaque ligne donne pour la station indiquée dans la première colonne les stations avec lesquelles son coefficient de corrélation est supérieur au seuil fixé, en ordre décroissant de coefficients de corrélation. Un '0' indique qu'il n'y pas plus d'autre station « bien » corrélée.

c. Calcul du NEVD optimal

Pour effectuer la reconstruction d'une station X à l'aide d'une station Y, nous cherchons à connaître le nombre de valeurs distinctes optimal associé à ce couple et donc à cette reconstruction. Pour cela nous procédons suivant cet algorithme :

```
Extraire les données de X et Y sur leur intervalle commun

Séparer les vecteurs obtenus en deux périodes

Calculer le nombre de valeurs distinctes de la première période de Y

Pour chaque NEVD possible :

      Diviser la première période de Y en intervalles associés au NEVD

      Construire la table des espérances conditionnelles de la première
      période de X en fonction de ces intervalles

      Reconstruire la seconde période de X à l'aide de la table des espé-
      rances conditionnelles

      Calculer l'indicateur de proximité

Trier les indicateurs de proximité en ordre croissant et garder le plus
petit. Le NEVD optimal est celui qui lui est associé.
```

Pour réaliser ces différentes étapes, les algorithmes suivants sont nécessaires.

i. Extraction des données communes

L'algorithme appliqué est le suivant :

```
Parcourir l'ensemble du vecteur X ;

Si la valeur de X existe :

      Regarder si la valeur de Y existe aussi

      Si X et Y existent, stocker les valeurs dans un tableau

Renvoyer le tableau des valeurs de X et Y sur leur intervalle commun
```

ii. Calcul des valeurs distinctes

Pour extraire les valeurs distinctes de Y_{C1}, la méthode mise en place est :

```
Trier le vecteur en ordre croissant,

Supprimer les doublons,
Compter le nombre de valeurs du nouveau vecteur ainsi obtenu,

Renvoyer le vecteur des valeurs distinctes et leur nombre.
```

Les étapes suivantes sont effectuées pour chaque NEVD possible.

iii. Séparation d'un vecteur en intervalles associés à un NEVD

Le NEVD est le nombre de valeurs distinctes par intervalle ; dans notre cas, nous cherchons à obtenir les bornes des intervalles associés à ce nombre.

En notant *k_nvd* le NEVD considéré, nous procédons de la façon suivante :

```
Stocker la première valeur du vecteur des valeurs distinctes,

Toutes les k_nvd composantes du vecteur des valeurs distinctes stocker la
valeur,

Renvoyer le vecteur des extrémités des bornes.
```

iv. Espérance conditionnelle d'un vecteur X en fonction d'un vecteur Y

Une fois les intervalles obtenus, nous construisons la table des espérances conditionnelles de X_{C1} en fonction de Y_{C1}. L'algorithme est :

```
Pour chaque intervalle de Y :

    Repérer les valeurs de Y qui en font partie, et les compter

    Sommer les valeurs de X correspondantes

    Normaliser la somme

Stocker les résultats dans la table des espérances conditionnelles de X
en fonction de Y.
```

Comme expliqué précédemment, la table des espérances conditionnelles de X en fonction de Y se présente de la manière suivante :

Intervalle de Y	Espérance de X
$\left[a(1), a(2)\right[$	$b(1)$
$\left[a(2), a(3)\right[$	$b(2)$
$\left[a(3), a(4)\right[$	$b(3)$
Etc.	Etc.

v. Reconstruction du vecteur X à partir de la table des espérances conditionnelles

A partir de la table des espérances conditionnelles il est possible de reconstruire la seconde période du vecteur X, X_{C2}.

La méthode est la suivante :

```
Pour chaque valeur de la seconde période de X,

    Regarder à quel intervalle appartient la valeur qui lui est asso-
    ciée dans Y,
```

```
        Remplacer dans la seconde période du vecteur X par la valeur donnée
        dans la table des espérances conditionnelles,

Renvoyer le vecteur reconstruit.
```

Le calcul de l'indicateur de proximité se fait à l'aide de la formule donnée plus haut. Une fois cet indicateur calculé pour chaque NEVD, nous comparons les valeurs obtenus et isolons la plus petite qui correspond au NEVD optimal.

Nous connaissons donc désormais le nombre de valeurs distinctes souhaitées dans chaque intervalle de Y.

Les étapes suivantes, c'est-à-dire la construction d'intervalles à partir d'un nombre de valeurs distinctes souhaitées par intervalle, la construction d'une table d'espérances conditionnelles et la reconstruction de données manquantes ont été décrites précédemment.

Nous ajoutons un algorithme supplémentaire, qui sélectionne les valeurs à reconstruire.

```
Parcourir les vecteurs X et Y,

Si valeur de X manquante :

        Vérifier si la valeur de Y correspondante est renseignée,

        Si c'est le cas, reconstruire si possible la valeur de X à l'ai de
        la table des espérances conditionnelles et remplacer dans le vec-
        teur.

        Sinon laisser la valeur de X non renseignée

Renvoyer le vecteur reconstruit.
```

C. Implémentation des algorithmes sous Matlab

a. Mise en forme préliminaires des données

Les données à reconstruire se présentent sous la forme d'un tableau Excel :

$donnees = (donnees)_{i,j}$: Elle contient les données de débits et se présente sous la forme :

	N° de la $1^{ère}$ station			N° de la i^{e} station			N° de la dernière station
$1^{ère}$ date				|			
				|			
				|			
j^{e} date	_____	_____	_____	Débit de la i^{e} station enregistré à la date j			
Dernière date							

Tableau 13 : Mise en forme des débits à reconstruire

La première colonne correspond aux dates, la première ligne aux stations. La case $(i+1\,,j+1)$ contient le débit de la station i enregistré à la date j.

REMARQUES :

- Cette matrice doit être entrée dans le premier onglet du fichier Excel.

- Pour le bon fonctionnement du programme, afin d'éviter des problèmes d'importation entre Matlab et Excel, les débits, les dates et les numéros de stations doivent être au format numérique et non au format texte. Il est donc indispensable de repérer les stations par un numéro et non pas leur nom.

- Le nombre de stations et de dates n'est pas limité a priori.

Après importation des données sous Matlab, nous avons fait le choix de stocker les noms des stations et les dates dans deux vecteurs distincts, et de simplifier la matrice de données en ne conservant que les valeurs de débits. Elle se présente donc sous cette forme :

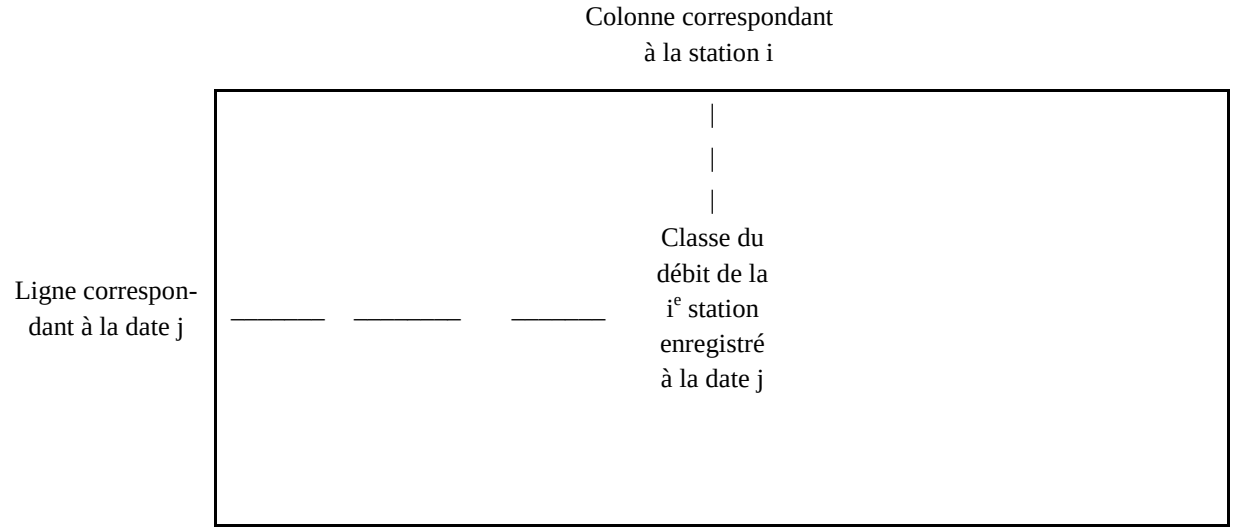


Tableau 14 : Présentation de la matrice donnees

Ce choix permet de simplifier les indices définissant les éléments de la matrice.
La case (i , j) contient désormais le débit de la station i enregistré à la date j.

b. *Programme principal de reconstruction des données manquantes :* `main.m`

Il s'agit du programme principal qui appelle les autres fonctions (en gras et soulignées) décrites dans la suite.

Le programme effectue les taches dans l'ordre suivant :

1. Lecture des débits des stations à étudier dans un fichier Excel ;
2. Mise en forme de ces données ;
3. Calcul des corrélations entre les stations ;
4. Détermination des stations les mieux corrélées ;
5. Reconstruction des données ;
6. Sortie des données reconstituées sous format Excel.

Le code Matlab correspondant est le suivant :

```
%------------- Reconstruction de données manquantes dans un tableau ------------

nb_decim=2;
nb_reconst=0;
nb_graph=0;
```

```
fileName = 'test_simple.xls';

donnees = xlsread(fileName);
taille=size(donnees);
nb_dates=taille(1)-1;
nb_stations=taille(2)-1;
stations = donnees(1,2:taille(2));
dates = donnees(2:taille(1),1);

donnees= donnees(2:taille(1),2:taille(2));
donnees = round(donnees*10^nb_decim)/10^nb_decim;

statistiques = stations';

for st = 1 : nb_stations
    statistiques(st,2) = length(donnees (isnan(donnees(1:nb_dates,st))==0, st)
);
end

tableau_correlations = correlation(donnees,stations,nb_dates,nb_stations);

xlswrite('donnees_reconstituees.xls',tableau_correlations,'correlations_stations
');

seuil_correl = input('Coefficient de corrélation minimum?') ;

while seuil_correl<0
    display('Ce coefficient doit être supérieur ou égal à 0')
    seuil_correl = input('Coefficient de corrélation minimum?');  % coeff de
corrélation minimum toléré
end

best_correl = extraction_correlation(tableau_correlations, nb_stations,
seuil_correl, stations);
nb_correls=size(best_correl,2);

for i=1:nb_stations
    for j=1:nb_correls
        best_correls(i,j+1)=best_correl(i,j);
        best_correls(i,1)=stations(i);
    end
end

best_correls

nb_figure=0;
toc
for st = 1:nb_stations
tic
    c = 1;
    colonne_X = stations(st);
    X = donnees(1:nb_dates,st);
    station_correlee = best_correl(st,1:nb_correls);
```

```
    while (size(X(isnan(X)==0),1) ~= length(X)) & (c <= nb_correls) & (sta-
tion_correlee(c) > 0)
        nb_reconst=nb_reconst+1;
        if ((nb_reconst-1)/4)==0+nb_figure
            figure
            nb_figure=nb_figure+1;
            nb_graph=0;
        end

        nb_graph=nb_graph+1;

        m=0;

        colonne_Y = station_correlee(c);

        for i=1:nb_stations
            if colonne_Y-stations(i)==0
                m=m+i;
            else m=m;
            end
        end

        Y = donnees(1:nb_dates,m);

        A = commun(X,Y);
        taille_XY=length(A);
        X_c = A(1:taille_XY,1);
        Y_c = A(1:taille_XY,2);


        X_c_1 = period(1,X_c);
        X_c_2 = period(2,X_c);

        Y_c_1 = period(1,Y_c);
        Y_c_2 = period(2,Y_c);

        clear A

        i_prox=0;

        for k_nvd = 1: nvd(Y_c_1)

            clear inter e_condi x_r

            inter = inter_nevd(Y_c_1,k_nvd);

            e_condi = esperance_condi(X_c_1 , Y_c_1 , inter);

            x_r=reconst(Y_c_2,e_condi);

            i_prox=i_prox+1;
            indprox(i_prox,1)=prox(x_r, X_c_2) ;
            indprox(i_prox,2)=k_nvd;
```

```
        end

        k_nvd;
        nevd_opti=sortrows(indprox,1);
        NEVD=nevd_opti(1,2);

        clear nevd_opti indprox x_r i_prox k_nvd

        inter=inter_nevd(Y_c,NEVD);

        max(Y_c);

        e_condi=esperance_condi(X_c,Y_c,inter);

        subplot(2,2,nb_graph);
        plot(e_condi(:,1),e_condi(:,2),'+');
        hold on;
        title(['station reconstruite: ',num2str(best_correls(st,1)),' - station
corrélée: ',num2str(station_correlee(c))])
        xlabel('valeur du débit de la 2e station');
        ylabel('espérance conditionnelle de la 1ere station');

        ch=choix_reconst(X,Y,e_condi);

        for i=1:nb_dates
            if (isnan(X(i))==1) & (isnan(ch(i))==0)
                X(i)=ch(i);
            end
        end


        c = c + 1 ;
        if c > length(station_correlee)
            break
        end

    end

    nouvelles_donnees(1:nb_dates,st) = X(1:nb_dates) ;
    toc

end
tic
for st = 1 : nb_stations
    statistiques(st,3) = length(nouvelles_donnees(isnan(nouvelles_donnees(:,st))
== 0, st) );
end

statistiques(:,4) = length(dates) - statistiques(:,3);
statistiques(:,5)= statistiques(:,3)-statistiques(:,2);

nouvelles_donnees;
nouvelles_donnees = [dates, nouvelles_donnees];
stations=[0, stations];
```

```
nouvelles_donnees = [stations; nouvelles_donnees];

disp('Statistiques sur le nombre de données reconstituées:')
disp(' ')
disp(statistiques)
disp(' colonne 1 : numéro de la station')
disp(' colonne 2 : nombre de données initiales')
disp(' colonne 3 : nombre de données après reconstruction')
disp(' colonne 4 : nombre de données encore manquantes')
disp(' colonne 5 : nombre de données reconstituées')

xlswrite('donnees_reconstituees.xls',nouvelles_donnees);
```

c. *Extraction des données communes :* `commun.m`

La reconstruction de données manquantes nécessite à plusieurs reprises les valeurs communes de deux vecteurs. Le but de cette fonction créée sous Matlab est donc d'isoler les intervalles d'intersections de deux vecteurs, et les valeurs qui en font partie.

La fonction reçoit en entrée un vecteur X et un vecteur Y, et renvoir une matrice des valeurs de X et de Y sur leurs intervalles communs.

```
function C=commun(x,y)

taille_C=0;


for i=1:length(x)
    if isnan(x(i))+ isnan(y(i)) == 0
        taille_C=taille_C+1 ;
        C(taille_C,1:2)=[x(i), y(i)];
    end
end
```

d. *Calcul des coefficients de corrélation :* `correlation.m`

Cette fonction permet de calculer le coefficient de corrélation entre deux jeux de données, à l'aide de la formule donnée précédemment.

Elle prend en entrée un tableau de données, les noms des stations et les dimensions du tableau. La fonction renvoie un tableau contenant les coefficients de corrélation des stations.

```
function  resultat =correlation(tableau,noms_stations,dim1,dim2)

count = 0;
j = 0;
num_correlation = 0;
nb_correlations = (dim2 - 1)*dim2;

resultat(1:nb_correlations,1:3) = 0;

for station = 1 : dim2

    X = tableau( isnan(tableau(1:dim1,station)) == 0 , station );
    X = X - mean2(X);


    for j = 1 : dim2

        if j ~= station

            a=tableau(1:dim1,station);
            b=tableau(1:dim1,j);

            AB=commun(a,b);
            taille_AB=size(AB,1);

            a = AB(1:taille_AB,1);
            b = AB(1:taille_AB,2);


            num_correlation = num_correlation + 1;

            resultat(num_correlation,  1)  =  noms_stations(station)  ;  %  numéro
station X
            resultat(num_correlation,  2)  =  noms_stations(j)  ;          %  numéro
station Y

            count = length(a);

            if count == 0
                resultat(num_correlation, 3) =  NaN ;
            else

                a = a - mean2(a);
                b = b - mean2(b);

                coef = sum(a.*b) / sqrt( sum(X.*X) * sum(b.*b) );

                resultat(num_correlation, 3) = coef;

            end

        end
    end
```

```
end
```

e. *Extraction des meilleures corrélations :* `extraction_correlation.m`

Cette fonction sert à sélectionner les coefficients de corrélation supérieurs à un certain seuil fixé.

Les paramètres d'entrée sont :

- le tableau des coefficients de corrélation ;
- le nombre de stations ;
- le seuil minimal ;
- les noms des stations.

La fonction renvoie un tableau recensant, pour chaque station, les corrélations considérées comme bonnes, classées par ordre décroissant de coefficient de corrélation.

```
function resultat = extraction_correlation(tableau, n, coeff_min, noms_stations)

for s = 1 : n

    clear liste
    nb_stations_correl = 0;

    tableau_s = tableau((tableau(1:size(tableau,1),3) > coeff_min ) & (tableau(1:size(tableau,1),3) ~= NaN ) & tableau(1:size(tableau,1),1) == noms_stations(s),:);
    nb_station_correl = size(tableau_s,1);

    if nb_station_correl ~= 0

        liste(1:nb_station_correl , 1) = tableau_s(1:nb_station_correl , 2);
        liste(1:nb_station_correl , 2) = tableau_s(1:nb_station_correl , 3);

        liste = sortrows(liste,2);

        for j=1:nb_station_correl
            resultat(s,j)=liste(nb_station_correl+1-j,1);
        end

    else
        resultat(s,1) = 0;

    end

end
```

f. Séparation de X_c et Y_c en périodes : `period.m`

Cette fonction permet de séparer un vecteur en deux périodes. Les données nécessaires sont un nombre *n*, valant 1 ou 2, selon la période de vecteur désirée, et le vecteur *X* à séparer. La fonction renvoie la période de *X* (première ou deuxième partie du vecteur, selon la valeur de *n*)

```
function P=period(n,x)

d=size(x,2);
k=size(x,1);

if n==1
    P=x(1:floor( k / 2 ),1:d);
elseif n==2
    P=x(floor( k / 2 )+1:k,1:d) .* (n==2) ;
else error('Vérifiez la période désirée (n=1 ou 2)');

end
```

g. Calcul du nombre de valeurs distinctes : `nvd.m`

Pour calculer le Nombre de Valeurs Distinctes (NVD) de Y_{C1}, nous utilisons une fonction existante de Matlab, *u*`nique.m`, qui trie les valeurs d'un vecteur dans l'ordre croissant en supprimant les doublons. Cette fonction n'éliminant pas les valeurs non renseignées, *NaN*, nous l'appliquons sur le vecteur Y_{C1} sans ces valeurs.

La fonction prend en entrée le vecteur *Y* dont on cherche à connaître le nombre de valeurs disctinctes.

```
function y=nvd(x)

y=length(unique(x(isnan(x)==0)));
```

h. Découpage d'un vecteur en intervalles : `inter_nevd.m`

La fonction Matlab créée à cet effet reçoit en entrée le vecteur à découper et le nombre de valeurs distinctes souhaité par intervalle. On obtient en sortie le vecteur des extrémités des intervalles.

```
function y=inter_nevd(x,n)

NEX= floor( (nvd(x)-1) / n ) + 1;

x_dis=unique(x(isnan(x)==0));

for i=0:NEX-1
    y(i+1)=x_dis(i*n + 1);
end
```

i. Calcul des espérances conditionnelles : `esperance_condi.m`

La fonction Matlab créée pour effectuer cette action reçoit en entrée :

- un vecteur X dont on cherche l'espérance conditionnelle ;
- un vecteur Y appelé vecteur de conditionnement ;
- un vecteur a contenant les bornes des intervalles définis sur Y.

Elle renvoie la table des espérances conditionnelles de X en fonction de Y.

```
function e = esperance_condi(vect_X,vect_Y, a )

a=[a,max(vect_Y)];

a=unique(a);

for i=1:length(a)-1
    z= vect_X .* ( vect_Y >= a(i) ) .* (vect_Y < a(i+1) );
    n=length( vect_Y(( vect_Y>=a(i) ) & ( vect_Y < a(i+1) ) ));
    e(i,1:2)=[sum(z) a(i)];

    if n>0
        e(i,1)= e(i,1)/n;
    else
        e(i,1)= NaN;
    end

end
```

j. Reconstruction d'un vecteur : `reconst.m`

Cette fonction créée sous Matlab est destinée à reconstruire un vecteur X en fonction d'un vecteur Y et de la table des espérances conditionnelles qui leur est associée.

Les données nécessaires sont :

- le vecteur Y ;
- la table des espérances conditionnelles associée aux vecteurs X et Y.

On obtient en sortie le vecteur X reconstruit.

```
function X2=reconst(Y,ec)

for i=1:length(Y)

    indice=0;

    for j=1:size(ec,1)
        indice=indice+ (Y(i)>=ec(j,2)) ;
    end
    if indice>0
        X2(i)=ec(indice,1)      ;
    else
        X2(i)=NaN;
    end
end

X2=X2';
```

k. Calcul de l'indicateur de proximité : `prox.m`

Cette fonction calcule l'indicateur de proximité d'une reconstruction, à partir du vecteur à reconstruire X et du vecteur reconstruit X^*.

```
function z=prox(x,x2)

a=(x-x2).^2 ;
b=x.^2;
if sum( b(isnan(a)==0 & isnan(b)==0)) == 0
    z=1;
else
    z=sqrt (   sum( a(isnan(a)==0 & isnan(b)==0) )     / ( sum( b(isnan(a)==0 &
isnan(b)==0) ) )   );
end
```

D. Mise en œuvre sur un exemple simple

Nous appliquons la méthode à un exemple simple constitué de 5 stations et de 20 dates. Les débits relevés sont reportés dans le tableau ci-dessous.

DATE	Station 1000	Station 2000	Station 3000	Station 4000	Station 5000
jour 1	2	3,3	2	2,5	2,1
jour 2	3	NaN	3,1	3,325	3,09
jour 3	4	NaN	NaN	1	0,3
jour 4	2,3	3,75	2	2,5	2,1
jour 5	4,2	6,6	4	4	NaN
jour 6	6,2	9,6	6	5,5	5,7
jour 7	1,56	2,64	2	2,5	2,1
jour 8	6,3	9,75	3	3,25	3
jour 9	4,3	6,75	4	4	3,9
jour 10	NaN	NaN	3	3,25	3
jour 11	NaN	6,3	3,9	3,925	3,81
jour 12	6,2	9,6	6,1	NaN	5,79
jour 13	2,6	4,2	3	NaN	3
jour 14	5,4	8,4	6	NaN	5,7
jour 15	2	3,3	2,3	2,725	2,37

Tableau 15 : Exemple d'application de la méthode de reconstruction des données manquantes

a. Calcul des corrélations

Les coefficients de corrélations $\rho_X(Y)$ sont :

X	Y	$\rho_X(Y)$
1000	**2000**	**0.9891**
1000	**3000**	**0.8304**
1000	4000	0.4761
1000	**5000**	**0.6786**
2000	**1000**	**0.9999**
2000	**3000**	**0.8286**
2000	**4000**	**0.6704**
2000	**5000**	**0.8272**
3000	**1000**	**0.8240**
3000	**2000**	**0.8184**

X	Y	$\rho_X(Y)$
3000	**4000**	**0.7166**
3000	**5000**	**0.9970**
4000	1000	0.5484
4000	**2000**	**0.6135**
4000	**3000**	**0.7818**
4000	**5000**	**0.9746**
5000	**1000**	**0.6761**
5000	**2000**	**0.6891**
5000	**3000**	**0.8415**
5000	**4000**	**0.7540**

Tableau 16 : Coefficient de corrélation des stations

On commence par déterminer les stations les mieux corrélées à chaque station. Nous nous fixons une limite acceptable minimum de 0.6 pour le coefficient de corrélation. Tous les coefficients du tableau ci-dessus n'étant pas supérieurs à 0.6, certains ne sont pas considérés comme « bons », et sont donc écartés.

Les stations les mieux corrélées pour chacune des stations sont :

	Station bien corrélée 1	Station bien corrélée 2	Station bien corrélée 3	Station bien corrélée 4
Station 1000	2000	3000	5000	0
Station 2000	1000	3000	5000	4000
Station 3000	5000	1000	2000	4000
Station 4000	5000	3000	2000	0
Station 5000	3000	4000	2000	1000

Tableau 17 : Corrélation des stations

Le tableau se lit comme suit : pour la station 1000, la station la plus corrélée est la 2000. Viennent ensuite les stations 3000, 5000 et 4000. Cela signifie que pour reconstruire la station 1000, on commence par utiliser la station 2000, puis s'il reste des données à reconstruire, on utilisera la station 3000…

b. Reconstruction de données manquantes (NaN)

Reconstruction de la station 1000 :

- Prise en compte de la station 2000 :

Dans ce cas, les résultats montrent que le NEVD optimal est 1. Cela signifie qu'il y a une valeur distincte par intervalle. Les bornes des intervalles sont donc les suivantes :

2.64 3.30 3.75 4.20 6.60 6.75 8.40 9.60 9.75

Les espérances conditionnelles de la station 1000 sachant que la station 2000 prend ses valeurs dans les intervalles ci-dessus sont :

Espérance conditionnelle de X	Intervalle de Y
1.56	[2.64 ; 3.30[
2.00	[3.30 ; 3.75[
2.30	[3.75 ; 4.20[
2.60	[**4.20** ; **6.60**[
4.20	[6.60 ; 6.75[
4.30	[6.75 ; 8.40[
5.40	[8.40 ; 9.60[
6.20	[9.60 ; +∞[

Tableau 16 : Espérance conditionnelle des débits de la station 1000 sachant ceux de la station 2000

Le tableau se lit comme suit : l'espérance de X sachant que la valeur de Y est comprise dans l'intervalle [2.64 ; 3.30[est 1.56.

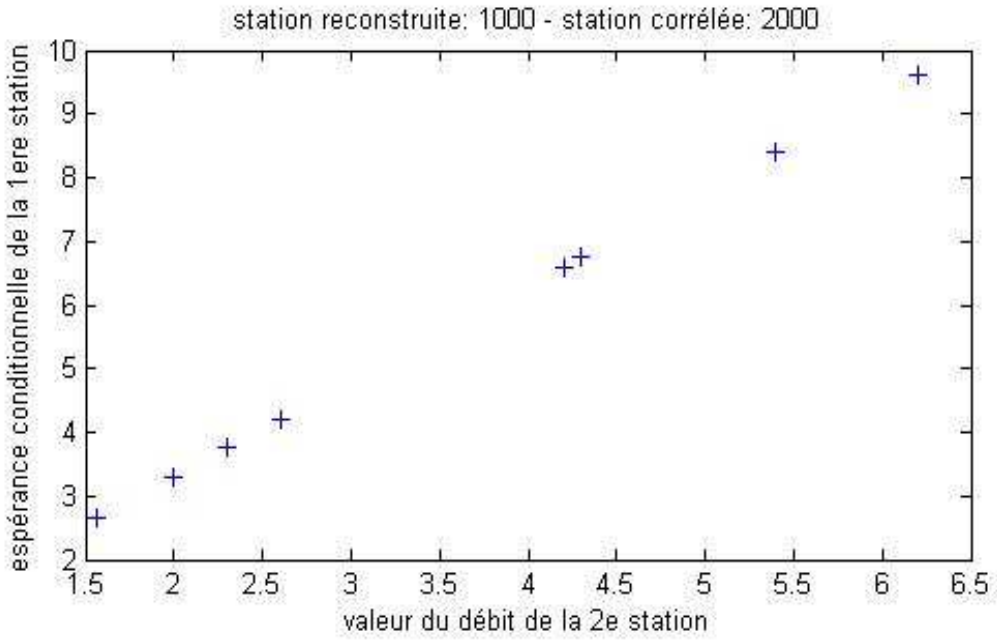


Figure 9 : Espérance conditionnelle des débits de la station 1000 sachant ceux de la station 2000

Le graphique présenté ci-dessus permet de visualiser la courbe de l'espérance conditionnelle de X en fonction du débit mesuré en Y. L'intérêt présenté par ce tableau est de permettre à l'utilisateur du programme de reconstruction de données de visualiser si une éventuelle valeur suspecte dans la table des espérances conditionnelles.

Nous souhaitons reconstituer le jour 10 et le jour 11 de X.

Au jour 11, on sait que Y a pris la valeur 6.3. Cette valeur est comprise dans l'intervalle [4.20 ; 6.60[. On affecte donc la valeur 2.60 à la donnée X du jour 11.

Au jour 10, il n'y a pas de donnée pour la station $Y = 2000$; on ne peut donc pas reconstruire la donnée manquante à partir de la station 2000. On passe donc à la deuxième station la plus corrélée.

- Prise en compte de la station 3000 :

Les résultats montrent que le NEVD optimal est 2. Cela signifie qu'il y a deux valeurs distinctes par intervalle. Les bornes des intervalles sont donc les suivantes :

2.00 3.00 3.90 6.00

Les espérances conditionnelles des débits de la station 1000 sachant ceux de la station 3000 prennent leurs valeurs dans les intervalles ci-dessus sont :

Espérance conditionnelle de X	**Intervalle de Y**
1.9650	[2.00 ; 3.00[
3.9667	[**3.00** ; **3.90**[
3.7000	[3.90; 6.00[
5.8000	[6.00 ; +∞[

Tableau 18 : Espérance conditionnelle des débits de la station 1000 sachant ceux de la station 3000

Le jour 10, on sait que Y a pris la valeur 3. Cette valeur est comprise dans l'intervalle [3.00 ; 3.10[. On affecte donc la valeur 3.9667 à la donnée X du jour 10.

Reconstruction de la station 2000 :

- Prise en compte de la station 1000 :

Les calculs montrent que le NEVD optimal est 1. Cela signifie qu'il y a une valeur distincte par intervalle. Les bornes des intervalles sont donc les suivantes :

1.56 2.00 2.30 2.60 4.20 4.30 5.40 6.20 6.30

Les espérances conditionnelles du débit de la station 2000 sachant que celui de la station 1000 prend ses valeurs dans les intervalles ci-dessus sont :

Espérance conditionnelle de X	Intervalle de Y
2.64	[1.56 ; 2.00[
3.30	[2.00 ; 2.30[
3.75	[2.30 ; 2.60[
4.20	[**2.60** ; **4.20**[
6.60	[4.20 ; 4.30[
6.75	[4.30; 5.40[
8.40	[5.40 ; 6.20[
9.60	[6.20 ; +∞[

Tableau 19 : Espérance conditionnelle des débits de la station 2000 sachant ceux de la station 1000

Nous souhaitons reconstituer les jours 2, 3 et 10 de X.

Le jour 2, on sait que Y a pris la valeur 3. Cette valeur est comprise dans l'intervalle [2.60 ; 4.20[. On affecte donc la valeur 4.20 à la donnée X du jour 2.

Le jour 3, Y a pris la valeur 4. Cette valeur est comprise dans l'intervalle [2.60 ; 4.20[. On affecte donc la valeur 4.20 à la donnée X du jour 3.

Au jour 10, il n'y a pas de donnée pour la station 1000, on ne peut donc reconstruire la valeur manquante à partir de la station 1000. On passe donc à la deuxième station la plus corrélée.

- Prise en compte de la station 3000 :

Le NEVD optimal est 2. Cela signifie qu'il y a deux valeurs distinctes par intervalle. Les bornes des intervalles sont donc les suivantes :

2.00 2.30 3.00 3.10 3.90 4.00 6.00 6.10

Les espérances conditionnelles du débit de la station 2000 sachant que celui de la station 3000 prend ses valeurs dans les intervalles ci-dessus sont :

Espérance conditionnelle de X	Intervalle de Y
3.230	[2.00 ; 2.30[
3.300	[2.30 ; 3.00[
6.975	[**3.00** ; **3.10**[
4.200	[3.10 ; 3.90[
6.300	[3.90 ; 4.00[
6.675	[4.00; 6.00[
9.000	[6.00 ; +∞[

Tableau 20 : Espérance conditionnelle des débits de la station 2000 sachant ceux de la station 3000

Le jour 10, Y a pris la valeur 3. Cette valeur est comprise dans l'intervalle [3.00 ; 3.10[. On affecte la valeur 6.975 à la donnée X du jour 10.

Reconstruction de la station 3000 :

- Prise en compte de la station 5000 :

Le NEVD optimal est 1. Cela signifie qu'il y a une valeur distincte par intervalle. Les bornes des intervalles sont donc les suivantes :

2.10 2.37 3.00 3.09 3.81 3.90 5.70 5.79

Les espérances conditionnelles de la station 3000 sachant que la station 5000 prend ses valeurs dans les intervalles ci-dessus sont :

Espérance conditionnelle de X	**Intervalle de Y**
2.00	[2.10 ; 2.37[
2.30	[2.37 ; 3.00[
3 .00	[3.00 ; 3.09[
3.10	[3.09 ; 3.81[
3.90	[3.81 ; 3.90[
4.00	[3.90 ; 5.70[
6.00	[5.70 ; +∞[

Tableau 21 : Espérance conditionnelle des débits de la station 3000 sachant ceux de la station 5000

Nous souhaitons reconstituer le jour 3 de X.

Pour ce jour, on sait que Y a pris la valeur 0.3. Cette valeur n'est comprise dans aucun intervalle de Y. On ne peut donc pas reconstituer la donnée manquante du troisième jour à partir de la station 5000.

- Prise en compte de la station 1000 :

Dans ce cas, le NEVD optimal est 1. Cela signifie qu'il y a une valeur distincte par intervalle. Les bornes des intervalles sont donc les suivantes :

1.56 2.30 3.00 4.30 6.20

Les espérances conditionnelles du débit de la station 3000 sachant que celui de la station 1000 prend ses valeurs dans les intervalles ci-dessus sont :

Espérance conditionnelle de X	Intervalle de Y
2.100	[1.56 ; 2.30[
2.500	[2.30 ; 3.00[
3.550	[**3.00** ; **4.30**[
5.000	[4.30 ; 6.20[
6.050	[6.20 ; +∞[

Tableau 22 : Espérance conditionnelle des débits de la station 3000 sachant ceux de la station 1000

Pour le jour 3, on sait que Y a pris la valeur 4. Cette valeur est comprise dans l'intervalle [3.00 ; 4.30[. On affecte donc la valeur 3.55 à la donnée X du jour 3.

Reconstruction de la station 4000 :

- Prise en compte de la station 5000 :

Dans ce cas, le NEVD optimal est 1. Cela signifie qu'il y a une valeur distincte par intervalle. Les bornes des intervalles sont donc les suivantes :

0.30 2.10 2.37 3.00 3.09 3.81 3.90 5.70

Les espérances conditionnelles du débit de la station 4000 sachant que celui de la station 5000 prend ses valeurs dans les intervalles ci-dessus sont :

Espérance conditionnelle de X	Intervalle de Y
1.00	[0.30 ; 2.10[
2.50	[2.10 ; 2.37[
2.73	[2.37 ; 3.00[
3.25	[**3.00** ; **3.09**[
3.33	[3.09 ; 3.81[
3.93	[3.81 ; 3.90[
4.00	[**3.90** ; +∞[

Tableau 23 : Espérance conditionnelle des débits de la station 4000 sachant ceux de la station 5000

Nous souhaitons reconstituer les jours 12, 13 et 14 de X.

Le jour 12, on sait que Y a pris la valeur 5.79. Cette valeur est comprise dans l'intervalle [5.70 ; +∞[. La valeur 4 est donc affectée à la donnée manquante du jour 12.

Au jour 13, on sait que Y a pris la valeur 3. Cette valeur est comprise dans l'intervalle [3.00 ; 3.09[. On affecte donc la valeur 3.25 à la donnée manquante du jour 13.

Au jour 14, on sait que Y a pris la valeur 5.7. Cette valeur est comprise dans l'intervalle [5.70 ; +∞[. De la même manière que pour le jour 12, la valeur 4 est affectée à la donnée manquante.

Reconstruction de la station 5000 :

- Prise en compte de la station 3000 :

Dans ce cas, le NEVD optimal est 1. Cela signifie qu'il y a une valeur distincte par intervalle. Les bornes des intervalles sont donc les suivantes :

2.00 2.30 3.00 3.10 3.90 4.00 6.00 6.10

Les espérances conditionnelles du débit de la station 5000 sachant que celui de la station 3000 prend ses valeurs dans les intervalles ci-dessus sont :

Espérance conditionnelle de X	**Intervalle de Y**
2.10	[2.00 ; 2.30[
2.37	[2.30 ; 3.00[
3.00	[3.00 ; 3.10[
3.09	[3.10 ; 3.90[
3.81	[3.90 ; 4.00[
3.90	[**4.00** ; **6.00**[
5.70	[6.00 ; +∞[

Tableau 24 : Espérance conditionnelle des débits de la station 5000 sachant ceux de la station 3000

On souhaite reconstruire la donnée du jour 5. On sait que ce jour là, Y a pris la valeur 4. Cette valeur est comprise dans l'intervalle [4.00 ; 6.00[. On affecte donc la valeur 3.90 à la donnée X du jour 5.

Résultats

Ainsi, étape par étape, nous avons reconstruit les données manquantes sur l'ensemble des stations :

DATE	Station 1000	Station 2000	Station 3000	Station 4000	Station 5000
jour 1	2	3,3	2	2,5	2,1
jour 2	3	**4.2**	3,1	3,325	3,09
jour 3	4	**4.2**	**3.55**	1	0,3
jour 4	2,3	3,75	2	2,5	2,1
jour 5	4,2	6,6	4	4	**3.90**
jour 6	6,2	9,6	6	5,5	5,7
jour 7	1,56	2,64	2	2,5	2,1
jour 8	6,3	9,75	3	3,25	3
jour 9	4,3	6,75	4	4	3,9
jour 10	**3.97**	**6.975**	3	3,25	3
jour 11	**2.6**	6,3	3,9	3,925	3,81
jour 12	6,2	9,6	6,1	**4**	5,79
jour 13	2,6	4,2	3	**3.25**	3
jour 14	5,4	8,4	6	**4**	5,7
jour 15	2	3,3	2,3	2,725	2,37

Tableau 25 : Relevés de débit des cinq stations de l'exemple après reconstruction des données manquantes

Dans cet exemple très simple, toutes les valeurs manquantes ont pu être reconstruites. Cela n'est pas forcément le cas pour tous les jeux de données. Tout dépend de la qualité des corrélations entre les différentes stations et du nombre de données manquantes.

E. Etude de l'évolution de l'indicateur de proximité

A la demande de la CCR, nous avons étudié l'évolution de l'indicateur de proximité en fonction du nombre de valeurs distinctes par intervalle.

L'objectif est de réduire le temps passé à la recherche du NVD optimal. Nous avons cherché à mettre en évidence un comportement particulier de l'indice de proximité, à partir des données dont nous disposons : il n'en existe pas.

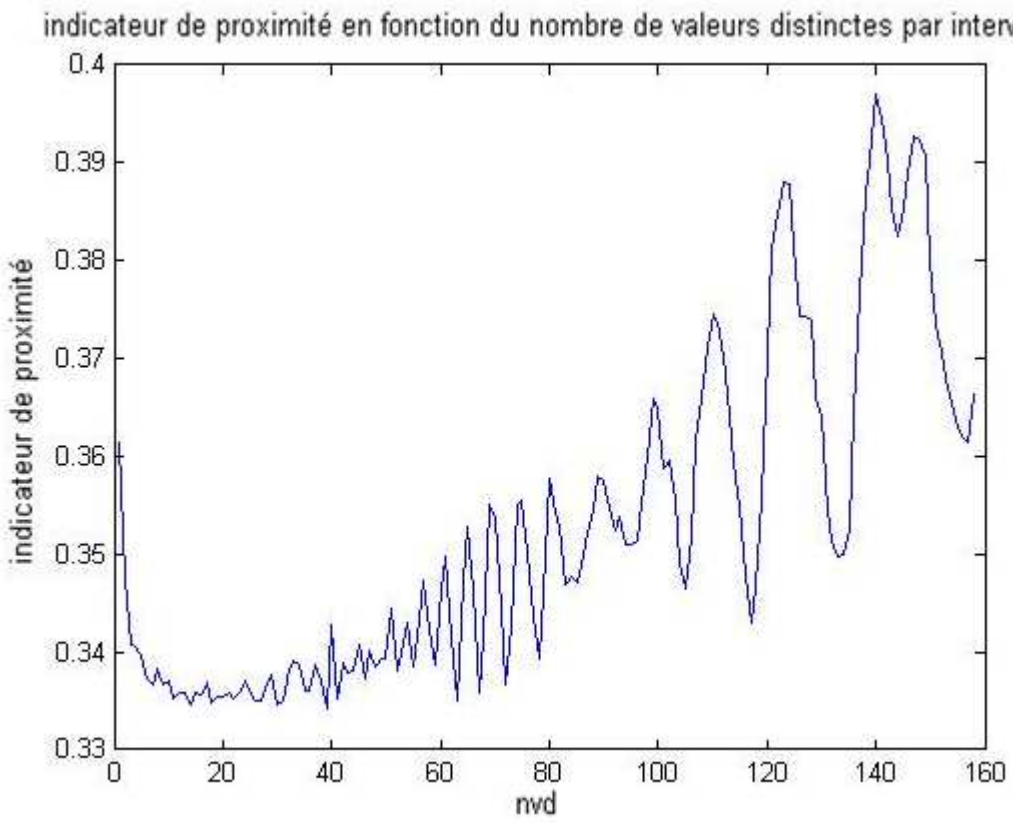


Figure 10 : Evolution de l'indicateur de proximité en fonction du nombre de valeurs distinctes par intervalle

Comme on le voit, la valeur de l'indice oscille sur toute la plage de NVD possibles. Il n'est donc pas possible de réduire le temps de calcul en ce qui concerne la recherche du NVD optimal : il est nécessaire de faire le calcul sur tous les NVD possibles.

V. Exemple d'application

A. Réflexion sur le nombre de stations à prendre en compte

Comme on l'a vu dans la partie II de ce rapport, le nombre de données manquantes de l'historique est particulièrement important :

- Si l'on commence l'historique à la date du premier relevé, toutes stations confondues (année 1936), le pourcentage de données manquantes s'élève à 66% ;

- Si l'on commence l'historique en 1970, 41% des données manquent.

Ces chiffres étant importants, il n'est pas forcément pertinent de reconstituer l'ensemble des données manquantes, pour utiliser ensuite une loi conjointe construite à partir de données reconstituées à près de 50%.

La CCR a émis plusieurs choix possibles pour l'application de la loi conjointe :

- Sur l'ensemble des stations. Cette option pose le problème que nous venons de soulever, à savoir un pourcentage de données manquantes important ;

- Sur une cinquantaine de stations, jugées bien relevées et représentatives de l'ensemble du bassin versant ;

- Sur 4 à 5 stations uniquement, ce qui pose le problème de ne représenter que des zones très localisées du bassin versant.

Le fait de travailler avec quelques stations ou bien avec l'ensemble des stations ne modifie pas les lois conjointes. Le fait d'ajouter des stations à la zone étudiée ne fait qu'ajouter des possibilités.

B. Application à quatre stations

Nous donnons ici un exemple d'utilisation des différents algorithmes et fonctions créés.

a. Exemple traité

Quatre stations ont été choisies sur le bassin versant de la Seine :

- H5071010 (Marne)

- H 0400010 (Seine à Bar sur Seine)

- H 7401010 (Oise à Sempigny)

- H 5920010 (Seine à Paris)

b. Données disponibles

Les quatre stations n'ont pas été toutes relevées de la même façon, comme on peut le voir dans le tableau ci-dessous :

	H5071010	**H0400010**	**H7401010**	**H5920010**
Date de premier relevé	01/01/1948	01/01/1950	01/01/1955	01/01/1974
Date de dernier relevé	11/12/2007	09/09/2007	07/06/2009	02/06/2009
Nombre de dates relevées	21796	21069	19879	12937
Début de l'historique pris en compte		01/01/1948		
Fin de l'historique		07/06/2009		
Nombre de données manquantes	643	1370	2560	9502
% données manquantes	3%	6%	11%	42%

Tableau 26 : Analyse statistique des données

Nous sélectionnons un historique débutant à la date de premier relevé la plus ancienne (début 1948) et s'achevant à la date de dernier relevé la plus récente (juin 2009). La plage d'étude est constituée de 22 440 jours.

On constate qu'il manque un certain nombre de données. La station la plus touchée est celle située sur la Seine à Paris, pour laquelle nous ne disposons de relevés qu'à partir de 1974.

La sélection des données et l'analyse statistique sont contenues dans le fichier Excel *SCM_CCR_Appli_4_stations_2009_09_30*.

c. Reconstruction données

Pour pallier l'absence de certaines données, nous utilisons l'algorithme de reconstruction de données manquantes. Il s'agit du programme Matlab *Reconstruction_main_prog*.

	H5071010	**H0400010**	**H7401010**	**H5920010**
Date de premier relevé	01/01/1948	01/01/1950	01/01/1955	01/01/1974
Date de dernier relevé	11/12/2007	09/09/2007	07/06/2009	02/06/2009
Nombre de dates relevées	21796	21069	19879	12937
Nombre de données après reconstruction	22439	22439	22439	22439
Nombre de données reconstituées	643	1370	2560	9502
Nombre de données encore manquantes	0	0	0	0
% données manquantes après reconstruction	0%	0%	0%	0%

Tableau 27 : Résultats de la reconstruction de données

L'algorithme permet de reconstituer toutes les données manquantes. Nous disposons donc d'un historique complet de 22 440 dates.

L'historique complet est contenu dans le fichier Excel *SCM_CCR_Donnees_reconstituees_2009_09_30*.

d. Loi conjointe

Nous construisons la loi conjointe des débits de ces quatre stations. Pour chacune nous définissons quatre classes, basées sur les débits biennaux, décennaux et cinquantennaux des stations.

Dans le cas de la station H5920010, située sur la Seine à Paris, le débit cinquantennal est supérieur au débit maximal observé depuis 1974 : $2\,200\,000\,l.s^{-1}$ pour le débit cinquantennal contre $1\,790\,000\,l.s^{-1}$ pour le débit maximal. Nous ne pouvons donc définir que trois classes sur cette station.

Les intervalles des classes sont donnés dans le tableau ci-dessous (en $l.s^{-1}$).

H5071010	**H400010**	**H7401010**	**H5920010**
440	175	3 000	19 000
160 000	120 000	110 000	1 100 000
270 000	210 000	190 000	1 600 000
360 000	280 000	260 000	2 200 000
496 000	315 000	278 000	

Tableau 28 : Définition des classes

Remarque : la valeur de débit maximal de la station située sur l'Oise (H7401010) ne correspond pas à celle calculée par la CCR. Mais cela n'a pas d'influence sur les résultats.

Le programme Matlab utilisé est : *loi_conjointe_main_prog*.

La loi conjointe des quatre stations contient 36 évènements de probabilité non nulle, dont 5 correspondent à une date unique (probabilité de 1/22440).

L'événement le plus probable correspond au cas où les quatre stations sont dans leur première classe (vecteur (1 1 1 1)) ; sa probabilité est de 0.96.

La loi conjointe est donnée dans le fichier *SCM_CCR_Loi_conjointe_2009_09_30*.

e. Exploitation de la loi conjointe

Nous utilisons à présent les différentes fonctions créées pour exploiter la loi conjointe.

i. *Calcul de la probabilité d'un évènement*

Nous cherchons la probabilité que les débits des quatre stations soient compris dans la classe extrême (débit supérieur au débit cinquantennal).

Pour cela, nous utilisons la fonction *p_evenement* : nous lui donnons en entrée le vecteur $(4, 4, 4, 3)$ et la matrice $proba$ contenant la loi conjointe des quatre stations. Le résultat est 0 : l'événement n'a jamais été observé. L'événement $(4, 4, 1, 2)$, quant à lui, a été observé au cours de l'historique : sa probabilité est de 4.46×10^{-5}.

La fonction *p_debit* permet de déterminer la probabilité d'un événement entré sous la forme de débits, et non pas de classes. Par exemple, nous lui donnons en entrée le vecteur $(333\,000\,, 275\,000, 253\,000, 1\,153\,000\,)$ et la matrice *proba*. La probabilité est 0 : ce quadruplet de débits n'a jamais été observé.

ii. *Tirage d'un n-uplet de débit*

La fonction Matlab *tirage* permet de simuler des évènements selon la loi conjointe. Elle renvoie un événement sous deux formes :

- Sous forme de n-uplet de classes ;
- Sous forme de n-uplet de valeurs de débit.

Les valeurs obtenues pour cet exemple sont les suivantes :

$$(1,1,1,1)$$

$$(185\,400, 174\,600, 148\,900, 1\,545\,600)$$

f. *Autres fonctions*

i. *Recherche d'un évènement*

La fonction *recherche* permet de rechercher et d'exporter dans un fichier Excel les dates et données de débit correspondant à un certain événement.

Nous l'avons utilisée pour rechercher dans l'historique les dates correspondant à l'événement $(4, 4, 1, 2)$. Il a été observé une fois, le 17 janvier 1955. Les débits des stations correspondants sont les suivants (en l/s) :

Dates	**H5071010**	**H400010**	**H7401010**	**H5920010**
19550117	372000	315000	50500	1121800

Tableau 29 : Date correspondant à l'événement $(4,4,1,2)$

ii. Calcul de probabilité pour une seule station

La fonction *probas_station_unique_main* permet de calculer les lois marginales des débits de chaque station. Elle calcule la probabilité que le débit d'une station donnée appartienne à une certaine classe.

Nous l'avons utilisée pour déterminer la loi du débit de la station H5071010 :

Classe	Probabilité
1	0.99
2	0.008
3	7×10^{-4}
4	2×10^{-4}

Tableau 30 : Loi marginale de la station H5071010

g. Utilisation des probabilités conditionnelles

Nous avons ensuite utilisé les probabilités conditionnelles pour reconstruire la loi conjointe lorsque chacune des quatre stations est dans sa classe extrême (débit supérieur au débit cinquantennal).

On suppose pour simplifier que la période de retour de la classe extrême est de 50 ans. On affecte la probabilité correspondante pour calculer les probabilités conditionnelles.

Prenons par exemple la station H5071010, située sur la Marne. La classe extrême des débits est l'intervalle $[360\, m^3 s^{-1}; 496\, m^3 s^{-1}]$ (classe n°4). On souhaite associer à cette classe la probabilité correspondant à une période de retour de 50 ans : cela signifie que tous les évènements tels que le débit de la station H5071010 est compris dans sa classe 4 doivent avoir une probabilité globale de 5.48×10^{-5}.

Pour ce faire, nous utilisons le programme *probas_conditionnelles_main*.

Les résultats pour cette station sont les suivants :

Probabilités avant valeur imposée sur station H5071010				
H5071010	H0400010	H7401010	H5920010	
4	2	1	2	8,9131E-05
4	3	1	2	8,9131E-05
4	4	1	2	4,4565E-05
Probabilités après valeur imposée sur station H5071010				
H5071010	H0400010	H7401010	H5920010	
4	2	1	2	2,1918E-05
4	3	1	2	2,1918E-05
4	4	1	2	1,0959E-05

Tableau 31 : Exemple d'utilisation des probabilités conditionnelles

Bibliographie

[1] Bernard Beauzamy : Robust mathematical methods for extremely rare events, preprint, 2009.

[2] Bernard Beauzamy et Olga Zeydina : Méthodes probabilistes pour la reconstruction de données manquantes. Ouvrage édité et commercialisé par la *Société de Calcul Mathématique S. A.,* ISBN : 2-9521458-2-2, ISSN : 1767 – 1175, avril 2007.

Annexe 1

Le tableau ci-dessous contient la liste des relevés manquants suite à des débits élevés. Les quatre lignes grisées correspondent à des dates spécifiques (31 décembre), ce qui nous laisse supposer que le manque de relevés est dû à une absence d'opérateur plus qu'à une crue. Les 23 autres peuvent être dus à des fortes crues.

Station		Date		Date
H5083050	du	05/01/1988	au	09/01/1988
	du	14/01/1995	au	01/02/1995
	du	07/04/1994	au	14/04/1994
H5321010	le	30/12/1993		
H6531010	du	12/11/2002	au	15/11/2002
H7102020	du	12/01/1993	au	15/01/1993
H2482010	du	21/12/1991	au	02/01/1992
	du	09/02/1994	au	15/02/1994
H6531011	le	09/11/1998		
H7742020	du	30/12/1999	au	31/12/1999
H9331010	du	03/11/1988	au	09/11/1988
	du	07/03/1989	au	23/03/1989
H5723010	du	26/01/1971	au	28/01/1971
H7313010	le	31/12/1994		
H7913020	du	13/02/1989	au	21/02/1989
	du	14/09/1999	au	24/09/1999
H5613020	du	31/12/1999	au	01/01/2000
H5723211	le	25/11/2000		
	le	27/11/2000		
	le	20/10/2001		
	le	22/10/2001		
	le	30/11/2001		
	le	03/12/2001		
	le	29/12/2001		
	le	31/12/2001		
	du	16/02/2006	au	28/02/2006
H7733010	du	08/12/1992	au	09/12/1992

Tableau 32 : Liste des relevés manquants après des débits élevés

Annexe 2

Les deux tableaux suivants contiennent la liste des relevés dont la valeur nulle nous a semblé suspecte.

Le premier tableau correspond aux relevés où le 0 affiché correspond à coup sûr à une donnée manquante. Ces valeurs ont été supprimées de l'historique et sont à reconstituer.

Le second tableau contient les relevés où le 0 affiché correspond peut-être à une donnée manquante. Dans le doute, nous n'avons pas modifié ces données.

Station		**Date**		**Date**	**Débit antérieur**	**Débit postérieur**
H2182010	du	19450719	au	19450719	1400,0	300,0
	du	19910825	au	19910825	200,0	500,0
	du	19910830	au	19910830	500,0	600,0
H6412010	du	19760705	au	19761223	50,0	50,0
H6201010	du	19790101	au	19790101	124000,0	101000,0
H6432010	du	19960809	au	19960812	14,0	45,2
H0800012	du	19970611	au	19970617	185000,0	264000,0
H5153010	du	20050712	au	20051204	0,8	368,0
	du	20041014	au	20041017	161,0	246,0
	du	20040512	au	20041009	27,6	820,0
	du	19860801	au	19860809	7,7	53,1
	du	19860906	au	19860911	1,9	74,1
	du	19950803	au	19950806	0,4	223,0
	du	19950819	au	19950821	0,0	77,2
	du	20050712	au	20051204	0,8	968,0
	du	20041014	au	20041017	161,0	246,0
	du	20040511	au	20041009	27,6	820,0
	du	20040226	au	20040508	aucune information	260,0
H7861010	du	20050510	au	20050510	aucune information	aucune information

Tableau 33 : Liste des débits nuls masquant des données manquantes

Station		Date		Date	Débit antérieur	Débit postérieur
H2622010	du	19760816	au	19761130	0,8	71,6
H6122010	du	19790807	au	19790807	9,0	18,0
H6412010	du	1960730	au	19961021	8,2	18,7
H6412020	du	19960811	au	19960812	0,1	56,6
H9403510	du	19760629	au	19760930	0,1	8,5
H7401010	du	20070626	au	20070702	18,8	0,0
H2001020	du	19900929	au	19900929	5,0	40,0
	du	19900918	au	19900921	1,0	10,0
H5153010	du	20041024	au	20041028	13,4	1,1
	du	19890823	au	19891103	0,3	28,7
	du	19900807	au	19900812	0,1	33,9
	du	19900820	au	19900829	0,2	5,5
	du	19911103	au	19911104	0,8	17,2
	du	19960819	au		0,0	11,6
	du	19910930	au	19911004	18,5	8,8
	du	20041024	au	20041027	13,4	1,1
	du	19960909	au	19960930	0,0	61,4
	du	19910702	au	19910705	0,2	46,9
	du	19910709	au	19910713	0,8	47,9
	du	19910716	au	19910729	0,1	19,5
	du	19910731	au	19910807	19,5	4,2
	du	19900820	au	19900829	0,2	5,5
	du	19911107	au	19911108	32,6	6,9
	du	19970609	au	19970610	24,1	95,0
	du	19970615	au	19970616	55,6	0,5
	du	19970624	au		27,7	1,6
H1603010	du	20030920	au	20030930	10,9	5,2
H3403201	du	19960921	au	19960930	0,1	17,4
	du	19960921	au	19960930	0,1	17,4
H1932020	du	19960801	au	19960812	26,4	9,9
	du	19760708	au	19761110	3,3	13,8
H1603010	du	19930827	au	19930924	1,4	10,7
H1713010	du	19930824	au	19930924	0,1	32,4
	du	20030920	au	20030930	10,9	5,2
H5726510	du	20030112	au	20030121	0,1	18,5
	du	20011015	au	20011019	0,0	11,2
	du	20000716	au	20000723	0,0	32,5
H5833010	du	19891012	au		0,3	23,1
H7061010	du	20070605	au	20070613	aucune information	0,0
	du	20070616	au	20070625	77,1	18,8

H5720010	du	19910612	au	19910723	aucune information	0,0

Tableau 34 : Liste des débits nuls masquant probablement des données manquantes