



HAL
open science

Modélisation des hydrosystèmes par approche systémique

Frédéric Bardolle

► **To cite this version:**

Frédéric Bardolle. Modélisation des hydrosystèmes par approche systémique. Sciences de la Terre. Université de Strasbourg, 2018. Français. NNT : 2018STRAH006 . tel-02002143

HAL Id: tel-02002143

<https://theses.hal.science/tel-02002143>

Submitted on 31 Jan 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**ÉCOLE DOCTORALE SCIENCES DE LA TERRE ET
ENVIRONNEMENT**

Laboratoire d'Hydrologie et de Géochimie de Strasbourg (LHyGeS)

THÈSE présentée par :
Frédéric BARDOLLE

soutenue le : **20 juin 2018**

pour obtenir le grade de : **Docteur de l'université de Strasbourg**

Spécialité : **Hydrogéologie quantitative**

**Modélisation des hydrosystèmes par
approche systémique**

THÈSE dirigée par :
D^r DELAY Frederick

Professeur, université de Strasbourg

RAPPORTEURS :
D^r FLIPO Nicolas
D^r MASSEI Nicolas

Maitre de Recherche, MINES ParisTech
Professeur, Université de Rouen Normandie

AUTRES MEMBRES DU JURY :

D^{re} DÖRFLIGER Nathalie

Directrice de la direction eau, environnement &
écotechnologies, BRGM

D^r POINOT Thierry

Professeur, Université de Poitiers

D^r POREL Gilles

Maître de conférences, Université de Poitiers

À Morgane.

SOMMAIRE

INTRODUCTION GÉNÉRALE	7
CONTEXTE DE L'ETUDE	7
OBJECTIFS DE LA THESE	7
PLAN DU MEMOIRE	8
CHAPITRE I : ÉLÉMENTS DE MODÉLISATION HYDROGÉOLOGIQUE	9
I.1 DEFINITION DES MODELES HYDROGEOLOGIQUES	9
I.2 PRINCIPE DE L'APPROCHE A BASE PHYSIQUE	10
I.3 PRINCIPE DE L'APPROCHE CONCEPTUELLE	13
I.4 PRINCIPE DE L'APPROCHE SYSTEMIQUE	16
I.4.1 Réseaux de neurones	17
I.4.2 Fonction de transfert	18
I.5 INVERSION DU MODELE	27
I.6 COMPLEXITE VERSUS SIMPLICITE, BIAIS VERSUS VARIANCE	29
CHAPITRE II : DESCRIPTION DU MODÈLE MASH	32
II.1 NON-LINEARITE DANS LES MODELES SYSTEMIQUES	32
II.1.1 Noyaux de Volterra	33
II.1.2 Fonction de production	34
II.1.3 Seuillage des entrées	35
II.2 ARCHITECTURE ET VERSATILITE DU MODELE	36
II.2.1 Versatilité du modèle	37
II.2.2 Architecture en série et en parallèle	39
II.2.3 Types de mise en relation possibles	41
II.3 FONCTIONS DE TRANSFERT	42
II.3.1 Courbes de récession	42
II.3.2 Courbes de récession des aquifères karstiques	47
II.3.3 Fonctions composites	49
II.3.4 Fonctions de densité de probabilités	49
II.3.5 Bibliothèque de fonction de MASH	51
II.4 INVERSION	56
II.4.1 PEST (Parameter ESTimation)	57
II.4.2 PSO (Particle Swarm Optimization)	66
CHAPITRE III : SIMULATION DE DONNÉES PROVENANT D'UN MODÈLE À BASE PHYSIQUE : LE MODÈLE MARTHE	74
III.1 DESCRIPTION DU MODELE A BASE PHYSIQUE	74

III.1.1	Calcul des charges des nappes	74
III.1.2	Calcul des débits des rivières	76
III.2	DEFINITION DES PARAMETRES DE L'EXPERIENCE	77
III.3	REPRODUCTION DU DEBIT A L'EXUTOIRE	81
III.3.1	Modèles à une zone	81
III.3.2	Modèles à deux, trois et quatre zones	83
III.4	CONTRIBUTION DES DIFFERENTES ZONES AU DEBIT ET FONCTIONS DE TRANSFERT	87
III.4.1	Modèles à deux branches	89
III.4.2	Modèles à trois branches	93
III.4.3	Modèles à quatre branches	96
III.5	CONCLUSIONS POUR CET EXEMPLE	99
CHAPITRE IV : SIMULATION DE DONNÉES PROVENANT D'UN CAS RÉEL : LE BASSIN VERSANT DE LA SÈVRE NIORTAISE À PONT-DE-RICOU		100
IV.1	PRESENTATION DE L'HYDROSYSTEME ETUDIE	100
IV.1.1	Géologie	102
IV.1.2	Hydrogéologie	102
IV.1.3	Données	103
IV.2	COMPARAISON DES ALGORITHMES D'OPTIMISATION STOCHASTIQUES	109
IV.2.1	Méthodologie	109
IV.2.2	Résultats	112
IV.2.3	Discussion des résultats	114
IV.3	COMPARAISON DES ARCHITECTURES DU MODELE	118
IV.3.1	Méthodologie	119
IV.3.2	Résultats	126
IV.3.3	Discussion des résultats	136
CONCLUSION GÉNÉRALE		140
ANNEXE A : DESCRIPTION DE L'INTERFACE GRAPHIQUE DE MASH		144
A.1	LES ENTREES	146
A.2	LA SORTIE	146
A.3	LES FONCTIONS DE TRANSFERT ET LEURS PARAMETRES	148
A.4	LES MENUS	150
ANNEXE B : COURBES SUPPLÉMENTAIRES RELATIVES AU CAS RÉEL		153
B.1	ÉVOLUTION DE L'AJUSTEMENT (R^2) POUR LES DIFFERENTS ALGORITHMES PSO	153
B.2	AJUSTEMENTS (NASH) POUR LES DIFFERENTES ARCHITECTURES DE MASH	166
RÉFÉRENCES BIBLIOGRAPHIQUES		181

REMERCIEMENTS

Je remercie chaleureusement M. Frederick Delay, directeur de thèse, pour son soutien sans faille.

J'adresse tous mes remerciements à MM. Flipo et Massei pour avoir accepté de rapporter ce travail et à M. Poinot pour avoir présidé le jury. Je remercie également les deux autres membres du jury, Mme Dörfliger et M. Porel avec une mention toute particulière pour leur dévouement et leurs conseils avisés alors qu'ils suivaient mes travaux entre 2010 et 2013.

Je tiens aussi à remercier mes collègues et mentors du BRGM Poitou-Charentes et du laboratoire HydrASA de l'Université de Poitiers.

Merci à ma famille et à mes amis pour leur support inconditionnel.

Enfin, ce travail n'aurait pas été possible sans le concours financier du BRGM, des agences de l'eau Adour-Garonne et Loire-Bretagne et de l'ex-Région Poitou-Charentes qui ont cofinancé le contrat doctoral de 2010 à 2013 durant lequel la majeure partie des travaux présentés ici ont été réalisés.



INTRODUCTION GÉNÉRALE

Contexte de l'étude

Les hydrosystèmes lorsqu'ils sont appréhendés dans leur ensemble pour des besoins de gestion – exploitation doivent intégrer des compartiments dont la dynamique est très contrastée. On citera pour exemple les contrastes de temps caractéristiques de transfert de l'eau entre : eaux de surfaces, écoulements hypodermiques, sols et nappes superficielles, nappes profondes. Dans l'état actuel des connaissances, il est impossible de poser correctement toute la physique et tous les besoins de changement d'échelles nécessaires afin de faire fonctionner ensemble dans un même modèle mécaniste les compartiments nommés ci-dessus. De plus, la construction de modèles tentant d'intégrer tous ces phénomènes physiques est très coûteuse, aussi bien en temps nécessaire à leur élaboration, qu'en données de conditionnement sur la géométrie du système, les propriétés des milieux, la maîtrise des conditions aux limites etc. Pour autant les pressions anthropiques accrues sur les hydrosystèmes supposent que leur gestion passe par une modélisation de qualité susceptible de donner des vues prospectives aux différents acteurs sociaux économiques actant sur la ressource en eau. Ce problème est d'autant plus contingent que la ressource en eau est soumise à exploitation intensive (par l'agriculture industrielle en l'occurrence pour la Région Poitou Charentes, mandataire de tout ou partie du travail présenté dans ce manuscrit).

Objectifs de la thèse

Les travaux présentés ici proposent de développer, confronter puis valider une approche systémique de ces milieux complexes. Dans le cas des hydrosystèmes, la dynamique est simplifiée en relations causales entre les différents flux d'échange du système. Les seules données nécessaires à la construction de modèles sont donc les différentes entrées et sorties du système. Grâce à cela, l'approche systémique propose de fournir un outil versatile qui soit simple de mise en place pour des systèmes dont on ne connaît pas pleinement le comportement. Pour s'assurer de cette versatilité, l'outil de modélisation développé dans ces travaux est éprouvé sur des données provenant d'un cas d'étude synthétique afin de vérifier sa capacité à s'adapter à des situations de complexité croissante. Ensuite, pour démontrer sa capacité à simuler des systèmes réels, le modèle est confronté à des données provenant d'un hydrosystème réel afin de s'assurer de la constance de ses performances en conditions concrètes d'application.

Au cœur du problème, hormis les aspects théoriques et mathématiques, se pose la question de l'inversion du modèle en lui conservant du sens physique tant pour ses paramètres que pour l'écriture des relations entre compartiments. Suivant les dernières avancées en apprentissage automatique (ou *machine learning*), la famille d'heuristiques choisie pour optimiser les paramètres du modèle est basée sur l'intelligence en essaim et se nomme « optimisation par essaim de particule » (ou PSO pour « *Particle Swarm Optimization* »). Différentes variations sont évaluées dans ces travaux en étudiant leur influence sur les capacités de simulation des modèles systémiques construits.

Plan du mémoire

Le mémoire est organisé en quatre chapitres principaux. Le premier chapitre introduit les éléments de modélisation hydrogéologiques ayant conduit à la création de l'outil de modélisation par approche systémique présenté dans ces travaux. Y sont décrites les trois grandes approches de modélisations hydrogéologiques (physique, conceptuelle et systémique), leurs avantages et leurs limites, ainsi que le concept d'inversion et le dilemme biais-variance. Le second chapitre détaille le modèle MASH (Modélisation des Hydrosystèmes par Approche Systémique), en particulier l'introduction des non-linéarités, la description des différentes architectures en série et en parallèle, le choix des fonctions de transfert utilisées et la comparaison de deux méthodes d'inversion, à savoir : PEST (« *Parameter ESTimation* ») et PSO (« *Particle Swarm Optimization* »). Le troisième chapitre présente un exemple d'application sur un cas d'école synthétique, établi à l'aide d'un modèle à base physique recréant un aquifère simplifié composé d'une à quatre zones ayant des caractéristiques hydrodynamiques différentes. L'objectif de ce test est de vérifier que MASH est capable de reproduire le comportement de systèmes (simplifiés) de complexité croissante et de mettre à l'épreuve sa capacité à identifier et à séparer les contributions au débit provenant de différentes zones d'un bassin versant ou d'une nappe à l'aide de fonctions de transfert disposées en parallèle. Le quatrième chapitre est dédié à un autre exemple d'application, cette fois sur un cas d'application réel. Il s'agit dans un premier temps de comparer les différents algorithmes d'optimisation afin d'évaluer leur capacité d'exploration et de convergence. Dans un second temps, l'étude porte sur la comparaison des différentes architectures qu'autorisent le modèle, et plus précisément sur l'apport éventuel des mises en série et en parallèle des fonctions de transfert, en validation croisée.

CHAPITRE I : ÉLÉMENTS DE MODÉLISATION HYDROGÉOLOGIQUE

I.1 Définition des modèles hydrogéologiques

Le terme « modèle » (de l'italien *modello*, du latin *modulus*, signifiant mesure) renvoie à l'origine aux représentations à échelle réduite. Il désigne ensuite ce qui est donné pour servir de référence puis prend une connotation plus scientifique au milieu du XX^{ème} siècle, avec l'apparition des termes dérivés « modéliser » et « modélisation ». L'évolution des modèles hydrogéologiques suit la même direction : des premières maquettes reproduisant des aquifères en modèles réduits, les scientifiques sont passés à l'utilisation de logiciels informatiques grâce à l'évolution fulgurante de la vitesse de calcul des micro-ordinateurs depuis les années 1970.

D'après le Dictionnaire français d'Hydrogéologie, la définition d'un « modèle hydrogéologique » est la suivante : « Toute représentation théorique simplifiée d'un système aquifère : elle comporte la définition d'une structure, la formulation de lois (relations déterministes ou probabilistes) reliant les impulsions ou entrées du système à ses réponses ou sorties, et la définition des paramètres (constantes ou variables pouvant prendre des valeurs numériques) entrant dans ces relations. [...] » (Figure I-1).

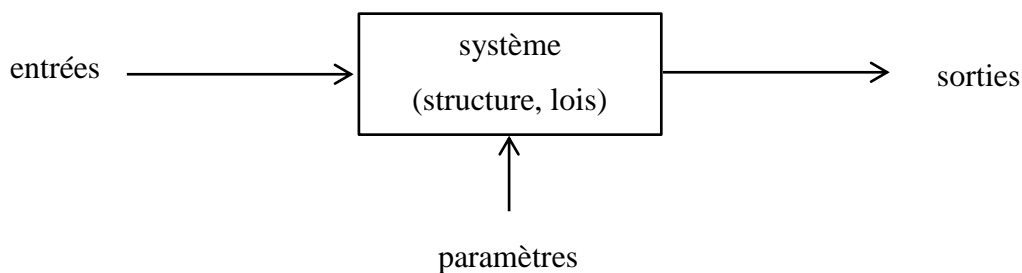


Figure I-1 : Schéma des composants d'un modèle hydrogéologique

Cette définition est suffisamment générale pour englober la majeure partie des très nombreux modèles hydrogéologiques existants. Chacun de ces modèles est plus ou moins adapté à un type de difficulté donnée, et sans vouloir entrer dans une procédure de hiérarchisation stipulant qu'un modèle puisse être meilleur que l'autre, on conviendra néanmoins qu'il n'existe pas de modèle apte à résoudre tous les problèmes. Plusieurs critères de classification des modèles hydrogéologiques cohabitent, comme leur appréhension de l'espace ou leur discrétisation temporelle. Certains auteurs [*Refsgaard and Storm, 1996; Ambroise, 1999*] classent les modèles, selon la manière dont ils représentent les processus, en trois catégories : les modèles à base physique, les modèles conceptuels et les modèles systémiques.

Les modèles à base physique ou mécaniste tentent de reproduire les caractéristiques du système (géométrie et propriétés physiques et géochimiques voir biologiques) de la manière la plus fidèle possible grâce un découpage de l'espace en mailles (1 à 3D) dont la taille et les paramètres

sont définis pour chaque problème. Dans chacune de ces mailles, des lois mathématiques décrivent les processus physiques majeurs considérés comme permettant de reproduire les observations avec un degré d'incertitude caractérisable. Les modèles conceptuels quant à eux simulent le fonctionnement macroscopique du système grâce à un ensemble de lois mathématiques relativement simples. En contexte hydrologique, le système simulé dispose d'un ou de plusieurs réservoirs interconnectés qui se vidangent les uns dans les autres, représentant les écoulements au niveau des différents compartiments du sol et sous-sol. Les processus physiques sont décrits de manière globale. Enfin, les modèles systémiques utilisent des méthodes purement numériques (comme le traitement du signal) afin de recréer un signal de sortie à partir d'un signal d'entrée. Les processus physiques peuvent être complètement ignorés, au profit du meilleur ajustement possible.

Plus les processus physiques sont représentés de manière fidèle par rapport à la réalité, plus le fonctionnement des modèles est complexe. Les modèles les plus complexes apportent une bonne compréhension du système mais nécessitent temps et expertise pour les mettre en œuvre et les utiliser. Au contraire, les modèles où les processus sont schématisés fonctionnent de manière plus simple. Ils sont rapides à créer et faciles à utiliser mais apportent peu de renseignements sur l'organisation et les mécanismes de fonctionnement du système. Le modèle développé au cours des travaux exposés ici se situe entre ces deux extrêmes : suffisamment complexe pour nous renseigner sur le fonctionnement du système mais suffisamment simple pour pouvoir être rapide et facile à utiliser. Les caractéristiques originales du modèle développé dans le cadre de cette thèse sont présentées à la lumière de l'état de l'art à la fin de ce chapitre, après une présentation des spécificités des modèles à base physique, conceptuelle et systémique.

I.2 Principe de l'approche à base physique

Les modèles à base physique, également appelés « maillés » ou « comportementaux », reposent sur la physique des écoulements. En effet, une des méthodes permettant de modéliser l'hydrodynamique des systèmes est la résolution des équations aux dérivées partielles issues de la physique des milieux continus. Mais comme la plupart du temps il n'existe pas de solution analytique à ces équations aux dérivées partielles, il est nécessaire de les discrétiser afin de pouvoir les résoudre numériquement. Cette discrétisation consiste à obtenir des valeurs numériques discrètes (i.e. en nombre fini) qui approchent la solution exacte grâce à trois méthodes principales (éléments finis, différences finis ou volumes finis) dont la pertinence dépend du type d'équation considérée.

De manière pragmatique, le terrain à modéliser est représenté par une grille (2D ou 3D) communément appelée maillage, et sur laquelle on résoudra une forme discrète des équations aux dérivées partielles susmentionnées. Les mailles de cette grille ont une forme géométrique donnée (e.g. carré, cubique, triangulaire ou prismatique) et contiennent des paramètres liés aux caractéristiques physiques du milieu concerné (e.g. la porosité, la conductivité hydraulique, le coefficient d'emmagasinement, ...). Ces paramètres sont mesurés directement sur le terrain ou extrapolés à partir de mailles proches et parfois ajustés a posteriori.

Dans l'optique de donner une solution au problème et de le rendre « calculable », des conditions aux limites du modèle sont définies à la périphérie de la zone modélisée. Les frontières du système, qu'elles soient naturelles ou purement géométriques définies par l'utilisateur, sont affectées de conditions classiques : type Neumann imposant un gradient de la variable d'état calculée (e.g. un flux pour un écoulement souterrain), ou de type Dirichlet imposant directement la valeur de la variable d'état (e.g. la charge hydraulique pour un écoulement souterrain). Une fois les paramètres des mailles et les conditions aux limites convenablement définis, le modèle calcule une solution numérique discrète de l'équation aux dérivées partielles, en régime permanent ou transitoire. Des séries de données d'observations expérimentales viennent alors améliorer le calage : le modèle est modifié de telle sorte que l'écart entre les données mesurées et leur équivalent simulé soit minimal. Pour ce faire, il est possible d'ajuster les paramètres des mailles, de jouer sur les conditions aux limites et les termes puits des équations, de raffiner le maillage dans des zones sensibles ou parfois de prendre en compte certains écoulements et autres processus préalablement négligés.

L'approche à base physique permet de suivre les réponses du modèle à la fois spatialement (grâce au maillage) et temporellement (grâce à la discrétisation temporelle en régime transitoire). Dans les autres approches (conceptuelle et systémique), le suivi spatial est au mieux limité, lorsque les modèles sont semi-distribués, au pire impossible, dans le cas des modèles globaux. À titre d'exemple, [Douez *et al.*, 2010] utilisent le modèle MARTHE (*Modeling Aquifers with Rectangular grids in Transitory regime for Hydrodynamic flow calculation*), développé par [Thiéry, 1993, 1994]) pour prendre en compte des prélèvements distribués géographiquement dans leur modèle de gestion de la ressource en eau (modèle « Jurassique » de la région Poitou-Charentes). Un des autres avantages majeurs de l'approche à base physique est la possibilité qu'elle offre d'imaginer des scénarios alternatifs et de simuler leur impact sur les réponses du système. Ainsi, dans leur modélisation du transport de nitrates dans un bassin versant, [Conan *et al.*, 2003] parviennent à simuler l'évolution de la quantité de nitrate présente dans la rivière dans le cas où l'épandage des effluents d'élevage dans les champs avoisinant diminuerait avec un des modèles les plus utilisés : MODFLOW (*Modular Three-Dimensional Ground-Water Flow Model*), développé par [MacDonald and Harbaugh, 1988]. Dans sa thèse, [Gomez, 2002] utilise une démarche similaire en évaluant les concentrations de nitrate pour différents scénarios de modification des pratiques agricoles avec le modèle MODCOU [Ledoux, 1980]. Dans un autre registre, [Cho *et al.*, 2009] simulent pour leur part l'impact d'un changement d'occupation des sols (ici l'urbanisation) sur le niveau des eaux souterraines et le débit des eaux de surface, ici aussi avec MODFLOW. Ce type de simulation de scénarios, basé sur la modification d'un ou plusieurs paramètres caractéristiques du terrain, même s'il est possible avec les autres approches, est grandement facilité par l'approche à base physique. En effet, dans cette dernière, les paramètres des mailles sont censés être directement des représentations des paramètres physiques du terrain (souvent plus ou moins homogénéisés néanmoins). Dès lors, il est relativement aisé de relier un scénario à un changement de paramètres du terrain, qui sera traduit pour le modèle en un changement d'un ou plusieurs paramètres des mailles. Pour les autres approches, établir un lien entre un changement des caractéristiques physiques du terrain et la modification de tel ou tel paramètre du modèle est beaucoup plus complexe. Il n'y a pas de « sens physique » immédiat entre une caractéristique

du milieu et les paramètres du modèle ; il faut souvent chercher ce lien et le comprendre à posteriori (i.e. au vu des résultats fournis par le modèle).

Il est également possible grâce à l'approche à base physique de tester des hypothèses quant au fonctionnement des systèmes aquifères. [Dewandel *et al.*, 2003] utilisent par exemple le modèle MARTHE pour vérifier dans quelles conditions les formules de Bousinesq et de Maillet permettent de retrouver certains paramètres physiques des aquifères. Étant donné que tous les paramètres du modèle sont connus, il est bien plus aisé d'affirmer ou d'infirmer de telles hypothèses grâce à un modèle à base physique, même complexe, qu'avec des données mesurées in situ sur un hydrosystème réel [Eisenlohr *et al.*, 1997].

En revanche, un des principaux inconvénients de l'approche à base physique est la difficulté à gérer de fortes hétérogénéités spatiales comme celles des systèmes karstiques [Fleury *et al.*, 2007]. Cela provient d'une géométrie des réseaux karstiques partiellement, voire totalement inconnue, alors qu'une connaissance exhaustive de cette dernière serait nécessaire afin de simuler correctement la complexité du système. Les expériences de traçage [Luhmann *et al.*, 2012] même si elles permettent de mettre en évidence la connectivité hydraulique des systèmes karstiques n'ont pas une résolution spatiale suffisante pour connaître la géométrie des drains. Toutefois, de nouvelles méthodes prometteuses voient le jour comme l'utilisation de sondes dotées d'accéléromètres réalisant une cartographie 3D des réseaux de conduits noyés [Pistre *et al.*, 2013] ou encore le recours à la modélisation génétique des réseaux karstiques [Lafare, 2011].

Au vu de la quantité de données nécessaire au bon fonctionnement des simulations ainsi que du travail nécessaire de la part du modélisateur pour créer un modèle « juste », l'approche à base physique est complexe. Tout d'abord, la quantité de données qu'il est nécessaire d'acquérir est très importante. En effet, chaque maille du modèle possède un certain nombre de paramètres liés aux caractéristiques physique du terrain et qu'il faut renseigner. Sauf exception, les mesures in situ ne couvrent pas la totalité des mailles du modèle distribué et il faut alors procéder par interpolation. Mais il est alors possible de passer à côté d'informations importantes, comme de fortes hétérogénéités spatiales de taille inférieure au pas de mesure. Ensuite, c'est l'approche qui demande le plus de travail lors de la création du modèle. Pour chaque nouveau système, le modélisateur doit recréer entièrement un nouveau modèle, et définir notamment quelle est sa géométrie, quelles données seront nécessaires pour définir les paramètres des mailles, etc. Enfin, cette complexité engendre des temps de calculs bien supérieurs à ceux des autres méthodes de modélisation. Étant donnée l'architecture générale des modèles à base physique, la masse des calculs nécessaires pour mener à bien une simulation augmente logiquement avec le nombre de mailles et le nombre d'itérations (pour les simulations en régime transitoire). Les hautes résolutions spatiales et temporelles sont par conséquent coûteuses en temps calcul et en paramètres (degrés de liberté) ce qui ne facilite pas la phase de calibration du modèle.

I.3 Principe de l'approche conceptuelle

Étant donné la complexité des hydrosystèmes et les difficultés inhérentes à la documentation puis la résolution précise des équations aux dérivées partielles, une autre approche s'est développée dans un souci de simplification. L'hydrosystème est alors représenté de manière globale et le cycle de l'eau est simplifié : l'eau provenant des précipitations est stockée temporairement dans différents réservoirs puis retourne soit dans l'atmosphère par évaporation soit dans les cours d'eau ou les nappes [Ambroise, 1999]. Dans cette approche, appelée « conceptuelle » ou « à réservoirs », l'hydrosystème est représenté comme un ensemble de réservoirs interconnectés par des conduits communicants. Le passage de l'eau à travers un réservoir ou un conduit est traduit dans le modèle par une équation mathématique simple (e.g. loi de vidange exponentielle pour un réservoir ou retard pour un conduit) [Singh, 1988].

Les modèles conceptuels, du fait de leur simplicité, sont parcimonieux ; en d'autres termes, ils utilisent peu de paramètres pour décrire le système. Le modèle GR4J de l'IRSTEA [Perrin *et al.*, 2003] possède par exemple quatre degrés de liberté et le modèle GARDENIA du BRGM [Thiéry, 1988] en comporte huit. De cette simplicité découle une rapidité des simulations car les calculs permettant de passer de l'entrée à la sortie sont simples et peu nombreux. Il est alors possible de rechercher les paramètres du modèle via des algorithmes d'optimisation avancés, par exemple de type évolutionnaire, qui nécessitent un grand nombre de simulations pour trouver un jeu de paramètres optimal. Attendu que les modèles issus d'une approche conceptuelle ne sont pas ou peu spatialisés, il n'est pas nécessaire d'acquérir de données fines concernant la géométrie du système, son hétérogénéité spatiale, etc. Les modèles sont donc plus simples et moins coûteux à créer que leurs homologues à base physique et donnent des résultats tangibles plus rapidement. Malgré cette simplicité, l'approche conceptuelle reste en partie ancrée dans la physique, contrairement aux modèles systémiques purement empiriques. Même si la complexité du fonctionnement des hydrosystèmes est largement gommée, le modèle conceptuel apporte une interprétation phénoménologique du fonctionnement du système, par exemple, la nappe se comporte comme un réservoir, qui se vidange progressivement dans une rivière, qui est elle-même un autre réservoir, etc. Quand bien même le fonctionnement du système serait représenté par des lois à base physique (e.g. vidange exponentielle) et que certains paramètres puissent renvoyer à une certaine réalité de terrain (e.g. la capacité de rétention d'un sol), le sens physique des modèles conceptuels n'est que partiel et il est fréquent que les paramètres d'ajustement ne puissent être confrontés (comparés) à une quelconque mesure réelle.

Cette simplification de la représentation et cette absence de spatialisation font que les modèles conceptuels, s'ils parviennent efficacement à reconstituer des séries de débits ou à générer des prévisions, sont en revanche mal adaptés à certains problèmes. Pour la simulation de scénarios alternatifs par exemple, quand le changement intervient sur les entrées (pluie ou évapotranspiration), ce qui est en partie le cas des simulations de l'impact du changement climatique, l'approche conceptuelle apporte des réponses. Mais quand les changements se produisent au niveau du système lui-même, comme une modification de l'occupation des sols, il est beaucoup plus difficile d'utiliser des modèles conceptuels car une modélisation des changements

implique de relier un ou plusieurs paramètres du modèle aux grandeurs connues mesurant les modifications. Dans la pratique, ce lien s'avère difficile à établir, mais pas impossible [Furusho *et al.*, 2013]. D'autres problèmes sont très sensibles à l'hétérogénéité du milieu, comme les phénomènes hyperboliques en général, (i.e., transport de masse, écoulement à surface libre, ...). S'il est question d'extraire d'autres informations que des temps caractéristiques, ces problèmes ne sont vraiment bien cernés qu'en spatialisant les équations locales des mécanismes et processus. Les modèles alors utilisés sont des modèles conceptuels semi-distribués comme EROS (*Écoulement dans une Rivière Organisée en Sous-bassins*) de [Thiéry and Moutzopoulos, 1992] ou TOPMODEL (*TOPography based hydrological MODEL*) de [Beven and Kirkby, 1979; Beven *et al.*, 1984].

Bien qu'il existe diverses architectures au sein de l'approche conceptuelle, les modèles comportent à minima deux réservoirs en série appelés fonction de production et fonction de routage [Charlier, 2007]. La fonction de production transforme les chroniques de pluie brute et d'évapotranspiration en chronique de pluie nette et la fonction de routage permet de passer de la chronique de pluie nette à la chronique de débit (**Figure I-2**). Cette fonction de routage peut ensuite se diviser en deux branches parallèles, avec d'une part, la réponse « lente » du système, représentant par exemple le soutien de la nappe au débit de base de la rivière, et d'autre part, la réponse « rapide » censée prendre en compte le ruissellement et autres écoulements à temps caractéristiques plutôt brefs. On retrouve ce type d'architecture pour la modélisation d'aquifères karstiques développée par [Fleury *et al.*, 2007; Moussu, 2011].

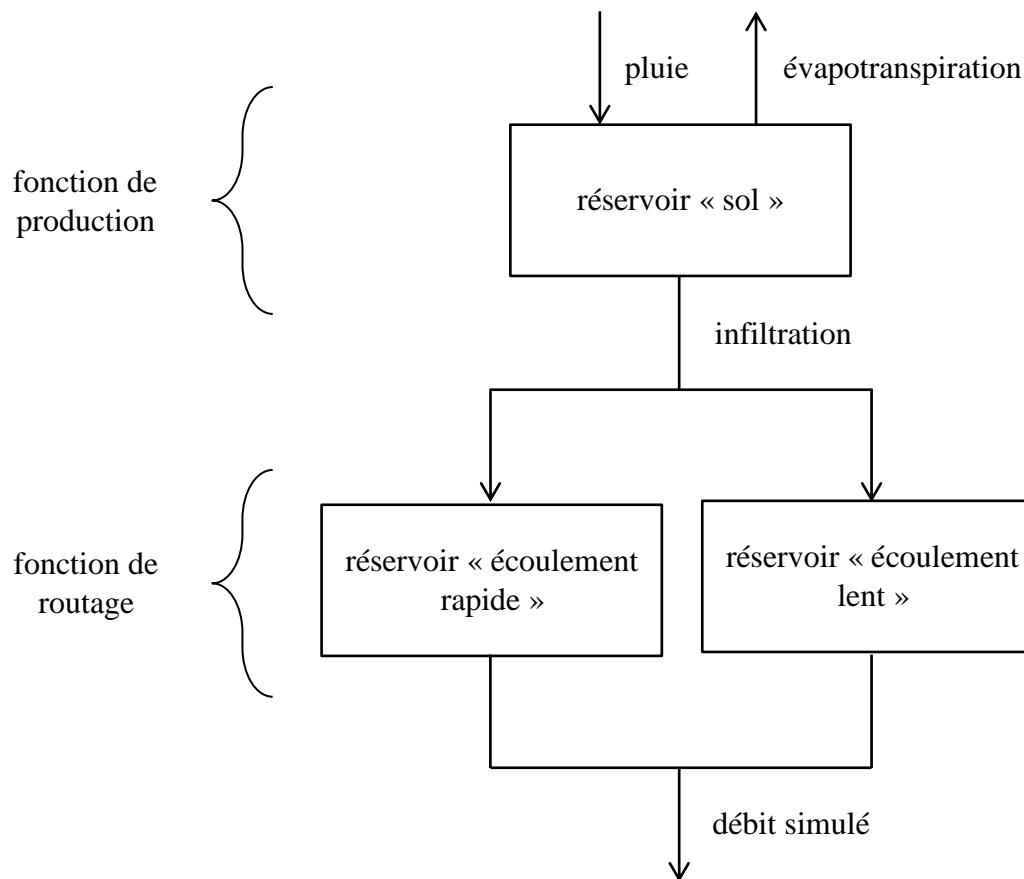


Figure I-2 : Schéma de fonctionnement d'un modèle conceptuel (simplifié d'après [Fleury et al., 2007])

Certains modèles ajoutent également des modules plus spécifiques à des contextes particuliers, comme la prise en compte de la fonte des neiges dans le modèle MORDOR développé par EDF [Paquet, 2004] ou la redistribution de la pluie par le couvert végétal en milieu tropical dans le modèle MHYDAS modifié par [Charlier, 2007]. Chaque modèle conceptuel a donc une structure qui lui est propre mais cette structure est rigide : le type d'entrée et de sortie qu'un modèle conceptuel peut manipuler ou encore le nombre de réservoirs mis en jeu sont en grande partie figés lors de la conception de l'outil. Si un modèle conceptuel est conçu pour n'utiliser que des chroniques de pluie et d'évapotranspiration pour modéliser un débit, l'intégration de données piézométriques mesurées reste impossible quand bien même ce renseignement puisse améliorer la compréhension du système et sa modélisation. À contrario, les modèles systémiques (voir plus bas), par exemple ceux basés sur les réseaux de neurones, sont capables d'utiliser en entrée n'importe quelles chroniques et ne sélectionnent que celles qui améliorent le calage de la simulation. Il y a donc un antagonisme certain entre la "rigidité" du modèle conceptuel fixant le principe de fonctionnement, le nombre, le type de ses réservoirs et le caractère universel d'un modèle dont on voudrait qu'il s'adapte à de nombreuses situations. Plutôt que d'intégrer de nombreux modules qui permettraient une adaptation à un grand nombre de situations mais ferait disparaître le caractère parcimonieux, les auteurs de modèles conceptuels acceptent que ceux-

ci ne soient utilisés que dans un nombre limité de cas. L'utilisation d'un modèle conceptuel doit donc se borner aux simulations de systèmes analogues à celui pour lequel le modèle a été conçu.

I.4 Principe de l'approche systémique

Les modèles systémiques diffèrent des modèles conceptuels par le fait qu'ils ne cherchent pas à décrire les processus impliqués dans la relation entrée-sortie et que leur structure n'est pas définie à l'avance [Bessiere, 2008]. Comme pour une approche conceptuelle, les paramètres du sol, la topographie du terrain ou la nature des aquifères sont ignorés. L'interprétation phénoménologique elle-même disparaît, au profit d'une représentation où le bassin versant (ou le bassin d'alimentation) est un système opaque ne possédant que des entrées et des sorties. La finalité de l'approche systémique, souvent qualifiée de « boîte noire », est de trouver la relation mathématique qui transforme les entrées en sorties. Cette relation caractérise alors l'hydrosystème et il devient possible de reconstituer des séries de données ou d'établir des prévisions sur les sorties comme avec les autres approches. La recherche d'une relation transformant des entrées en sorties peut être abordée via de nombreuses méthodes. Deux des méthodes les plus utilisées sont présentées ici : celles basées sur les réseaux de neurones, et celles utilisant des fonctions de transfert.

Les avantages et inconvénients des modèles systémiques et conceptuels sont globalement les mêmes : ils sont plus simples, parcimonieux et moins gourmands en données que leurs homologues à base physiques mais ne sont pas adaptés à la prévision si les caractéristiques physiques des bassins versants changent, et tiennent peu compte (pour les modèles semi-distribués) voire pas du tout (pour les modèles globaux) des variabilités spatiales. Il existe néanmoins des différences non négligeables entre ces deux approches. De par leur conception, les modèles systémiques n'ont pas la même rigidité que les modèles conceptuels. Les équations mathématiques permettant de passer de l'entrée à la sortie sont plus abstraites que les lois de vidanges utilisées dans les seconds et les modèles basés sur les réseaux de neurones ou les fonctions de transfert non-paramétriques peuvent comporter de nombreux degrés de liberté comme nous le verrons ultérieurement. Les modèles systémiques sont donc capables de s'adapter à une grande variété de problèmes parfois très différents. À titre d'exemple, un modèle comme TEMPO, développé par le BRGM, a pu être adopté pour traiter de questions telles que : l'évolution de l'oxygène dissous pour caractériser la résilience des écosystèmes aquatiques [Pinault and Berthier, 2007] ; le calculs des flux de nitrates afin de déterminer la capacité de dénitrification d'un aquifère [Pinault et al., 2001] ; la restitution de hauteurs piézométriques [Pinault and Schomburgk, 2006] et de chroniques de débits [Pinault et al., 2005]. Ceci est rendu possible par la versatilité des modèles systémiques, dont l'architecture interne est adaptable. À cela s'ajoute le fait que n'importe quelle chronique puisse être utilisée en entrée, permettant comme dans l'exemple évoqué avec TEMPO de "coupler" hydrogéologie et géochimie [Dörfliger et al., 2009]. Les modèles systémiques n'ayant pour seul objectif que la simulation au plus près des données mesurées, tout en restant libre de contraintes sur une quelconque physique à respecter, on constate généralement qu'ils restituent les chroniques de sortie plus fidèlement que les modèles conceptuels [Tokar and Markus, 2000] et les modèles à base physique [Frakes and Yu, 1999].

Le reproche le plus fréquemment adressé aux modèles systémiques est justement qu'ils ne reposent sur aucune base physique et n'apportent pas d'information sur le fonctionnement de l'hydrosystème. Certains auteurs essaient néanmoins d'attribuer du sens physique aux résultats des modèles systémiques pour tenter de justifier des comportements ou des valeurs de paramètres. [Jain *et al.*, 2004] montrent que les neurones de la couche cachée d'un modèle à réseau de neurones sont corrélés avec différents paramètres du système comme l'infiltration, le débit de base, le débit de surface, etc. [Moussa and Bocquillon, 1996] proposent d'utiliser comme fonction de transfert le noyau d'Hayami [Hayami, 1951] qui est une résolution de l'équation de l'onde diffusante. [Charlier *et al.*, 2008] l'utilisent afin d'acheminer une fraction de la pluie correspondant au ruissellement à l'exutoire du bassin versant. [Cudennec *et al.*, 2004] ont démontré qu'une fonction de transfert largement utilisée, la loi Gamma, pouvait être retrouvée comme fonction de densité de probabilité des temps de parcours à partir de la géomorphologie du bassin versant, en utilisant un raisonnement statistique sur les longueurs hydrauliques. Enfin, l'inversion des modèles systémiques reste un sujet épineux car les méthodes classiques d'optimisation, basées sur des calculs de gradients (dérivée d'une fonction d'écart modèle-données par rapport aux paramètres du modèle), se retrouve plus facilement piégées dans des minimums locaux de l'espace des paramètres [Chau, 2006]. Ces modèles engendrent en effet des problèmes plus « mal-posés » (*ill-posed*) que les modèles conceptuels, car ils ont plus de paramètres à optimiser, et de surcroît, ces paramètres sont moins contraints. Des algorithmes stochastiques sont donc généralement utilisés pour inverser les modèles.

I.4.1 Réseaux de neurones

Depuis une vingtaine d'années, les réseaux de neurones sont fréquemment utilisés dans le cadre de la modélisation pluie-débit [Shamseldin, 1997; Dawson and Wilby, 1998; Tokar and Johnson, 1999], avec comme exemple d'application la prédiction des crues éclair dans les systèmes karstiques [Johannet, 2011; Kong A Siou, 2011; Plantier Bornancin, 2013]. Dans ce genre de modèle, les neurones formels sont des analogues informatiques des neurones biologiques qui calculent une somme pondérée des variables qu'ils reçoivent en entrée (**Figure I-3**). Cette somme pondérée est seuillée par une fonction non-linéaire : si la somme est inférieure au seuil, la sortie vaut 0, le neurone est inactif ; si elle supérieure au seuil, la sortie vaut 1, le neurone est actif ; si elle se situe autour du seuil, elle suit la fonction (dite « d'activation ») généralement une sigmoïde ou une tangente hyperbolique. Les neurones sont ensuite organisés en réseau selon certaines architectures, la plus courante en hydrologie étant le perceptron multicouche [Jain *et al.*, 2004; Chau, 2006; Koffi *et al.*, 2007] (**Figure I-4**). Les neurones du réseau sont disposés en couches et sont reliés entre eux. Ils acceptent tous types d'informations en entrée et n'ont pas de schéma de fonctionnement prédéfini comme les modèles conceptuels. En effet, chaque neurone est activé s'il apporte des informations pertinentes au réseau. Les réseaux de neurones déterminent ainsi d'eux-mêmes quelle sont les chroniques nécessaires à la modélisation car le poids des entrées non-indispensables au modèle seront faibles après optimisation du problème inverse. Les paramètres à optimiser (i.e. les degrés de liberté) des modèles à réseaux de neurones sont donc les poids qui pondèrent les entrées des neurones. Il y a donc autant de paramètres qu'il y a de connexions entrée-neurone ou neurone-neurone.

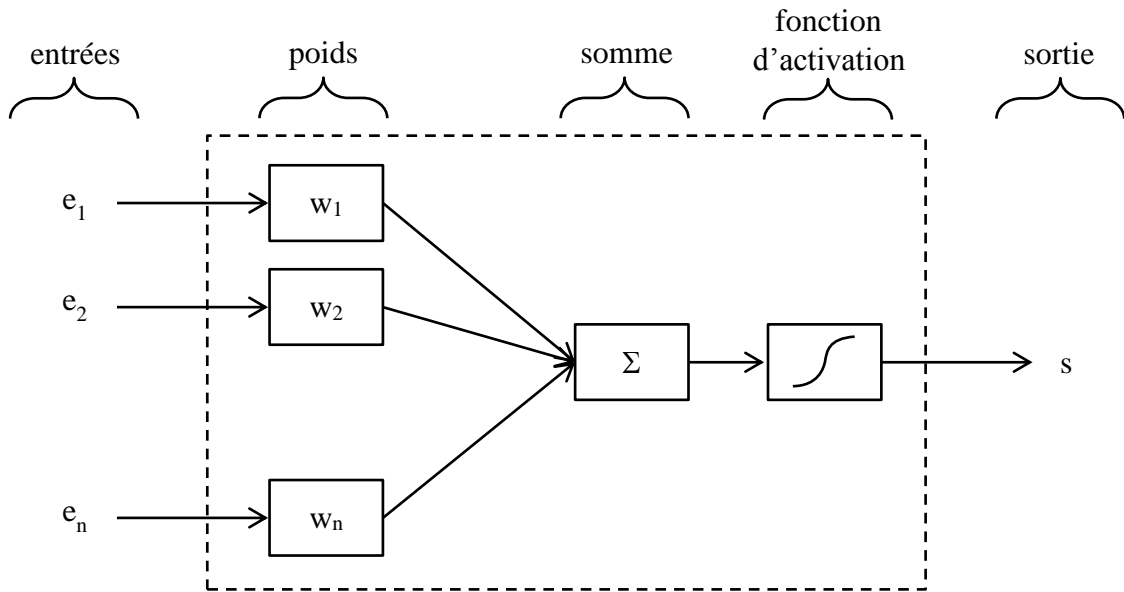


Figure I-3 : Principe de fonctionnement d'un neurone formel

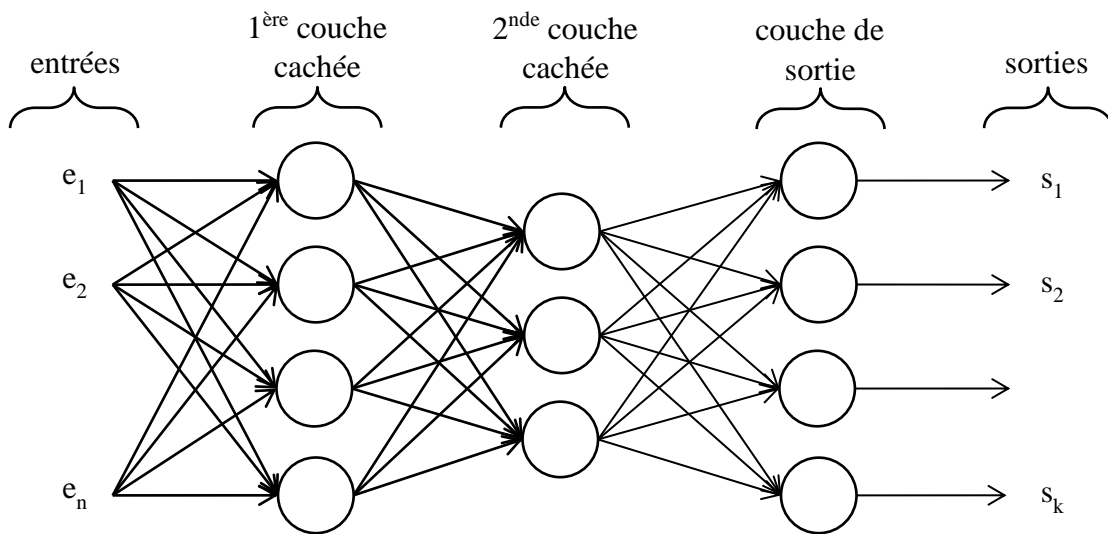


Figure I-4 : Principe de fonctionnement d'un réseau de neurones de type perceptron multicouche

I.4.2 Fonction de transfert

Une autre manière d'appréhender la modélisation des hydrosystèmes par l'approche systémique est d'utiliser des modèles basés sur les fonctions de transfert, ou noyaux de convolution. C'est cette approche qui a été utilisée lors des travaux présentés ici. Cette méthode, proche des théories issues du traitement du signal et de l'automatisme, consiste à établir une relation mathématique entre deux signaux, l'entrée et la sortie du système, via un produit de convolution. On peut interpréter le produit de convolution comme étant le filtrage du signal

d'entrée par une fonction de transfert, produisant ainsi un signal de sortie. De ce fait, on peut considérer que la fonction de transfert représente par exemple la manière dont le terrain filtre la pluie interceptée par le sol du bassin versant pour donner le débit de la rivière en sortie du bassin versant. La sortie du système $S(t)$ est égale au produit de convolution de l'entrée $E(t)$ et de la fonction de transfert $H(t)$ définie par :

$$S(t) = (E * H)(t) = \int_0^t E(\tau).H(t-\tau).d\tau \quad (\text{I.1})$$

soit de manière discrète :

$$S(n\Delta t) = \sum_{i \leq n} E((n-i)\Delta t)H(i\Delta t)\Delta t \quad (\text{I.2})$$

On peut voir le produit de convolution comme une généralisation de la moyenne mobile. Une des fonctions convoluée est transformée via une symétrie axiale selon l'axe des ordonnées puis est translatée le long de l'axe des abscisses. Le produit de convolution est alors égal, en chaque point de la translation $\tau \in]-\infty; \infty[$, à l'aire de l'intersection entre les deux fonctions (**Figure I-5**).

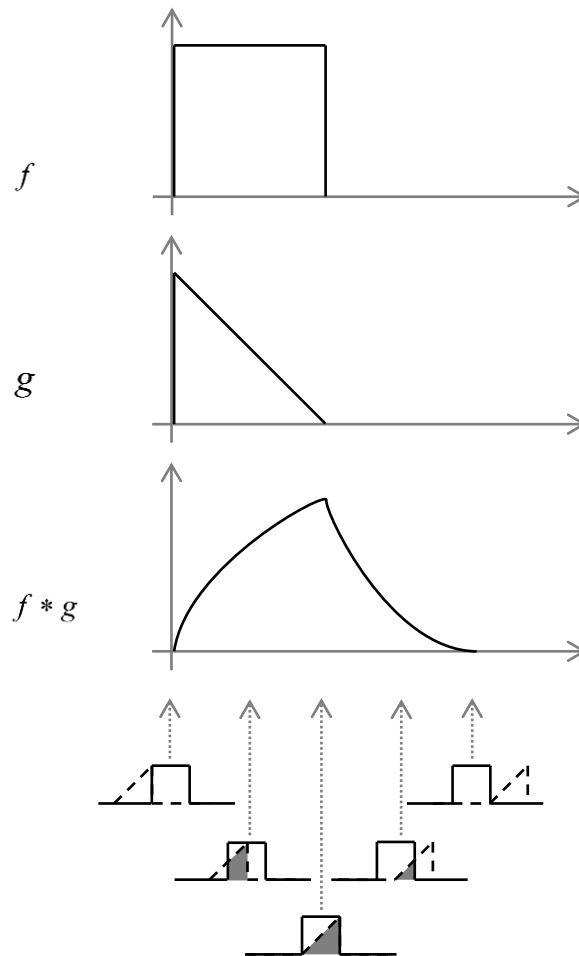


Figure I-5 : Représentation graphique du produit de convolution de deux fonctions f et g . En bas de la figure sont représentées les intersections au temps t entre $f(\tau)$ et la fonction $g(t-\tau)$ ayant subi une symétrie axiale et translaté de différents τ

Une autre manière de comprendre le produit de convolution est d'utiliser la notion de réponse impulsionnelle. Pour un système linéaire invariant, la réponse impulsionnelle est égale à la fonction de transfert. En effet, si l'entrée du système est une impulsion, c'est-à-dire une distribution de Dirac $\delta(t)$, définie par :

$$\delta(t) = \begin{cases} +\infty, & t = 0 \\ 0, & t \neq 0 \end{cases} \text{ et } \int \delta(t).dt = 1 \quad (\text{I.3})$$

alors la sortie $S(t)$ est égale à la fonction de transfert $H(t)$ car $\delta(t)$ est l'élément neutre de l'opération de convolution :

$$S(t) = (H * \delta)(t) = H(t) \quad (\text{I.4})$$

De manière discrète, la fonction de Dirac est définie plus simplement par :

$$\delta(t) = \begin{cases} 1, & t = 0 \\ 0, & t \neq 0 \end{cases} \quad (\text{I.5})$$

De la même manière, la fonction de Dirac translatée de a est définie ainsi :

$$\delta_a(t) = \begin{cases} 1, & t = a \\ 0, & t \neq a \end{cases} \quad (\text{I.6})$$

Considérons, dans le cas d'un bassin versant, une entrée définie par la chronique des pluies, la sortie définie par le débit d'une rivière. En isolant un évènement pluvieux de type impulsion, c'est-à-dire de durée inférieure ou égale au pas de temps d'échantillonnage, d'après l'équation (I.4) la réponse en débit du système serait égale à sa fonction de transfert, multiplié par l'intensité de l'impulsion (**Figure I-6**). Autrement dit, si $E(t) = K_1 \cdot \delta_1(t)$ avec K_1 l'intensité de la pluie à $t=1$, alors $S(t) = K_1 \cdot H(t)$.

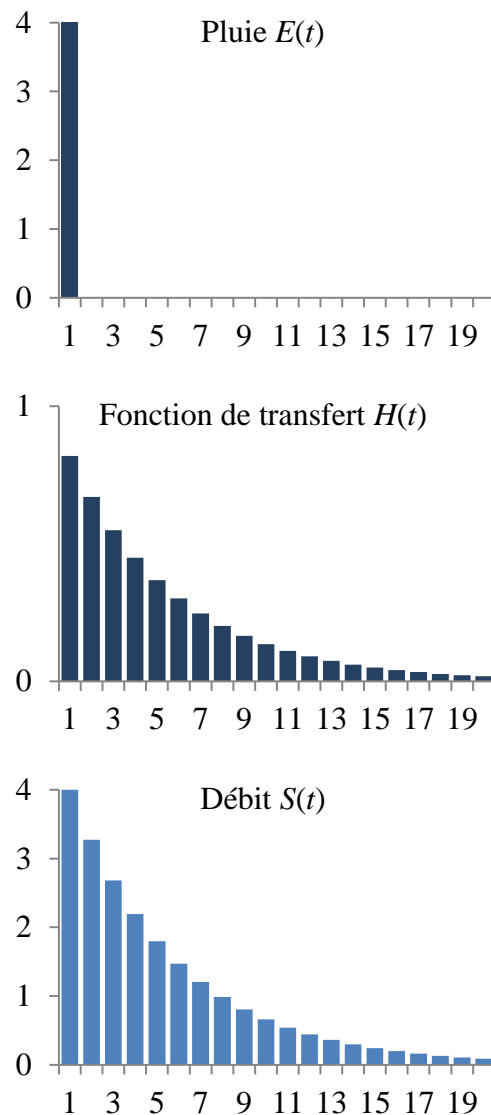


Figure I-6 : Impulsion (pluie), fonction de transfert et résultat du produit de convolution (débit)

Après avoir caractérisé le système, en ayant calculé sa réponse impulsionnelle, il est possible de calculer la valeur de la sortie au temps t à partir de la chronique d'entrée en faisant la somme de versions de la fonction de transfert translatées de i et multipliées par l'amplitude d'entrée de l'impulsion K_i au temps τ (**Figure I-7**). L'entrée étant égale à une somme de Dirac pondérés et translatés :

$$E(t) = \sum_{i=1}^{+\infty} K_i \cdot \delta_i(t) \quad (\text{I.7})$$

soit dans l'exemple présenté **Figure I-7**:

$$E(t) = K_1 \cdot \delta_1(t) + K_2 \cdot \delta_2(t) + K_3 \cdot \delta_3(t) \quad (\text{I.8})$$

Les valeurs de la chronique de sortie sont donc égales à :

$$\left\{ \begin{array}{l} S(1) = E(1) \cdot H(1) \\ \quad = K_1 \cdot \delta_1(1) \cdot H(1) \\ \quad = K_1 \cdot H(1) \\ S(2) = K_1 \cdot H(2) + K_2 \cdot H(1) \\ S(3) = K_3 \cdot H(1) + K_2 \cdot H(2) + K_1 \cdot H(3) \\ \dots \\ \text{etc.} \end{array} \right. \quad (\text{I.9})$$

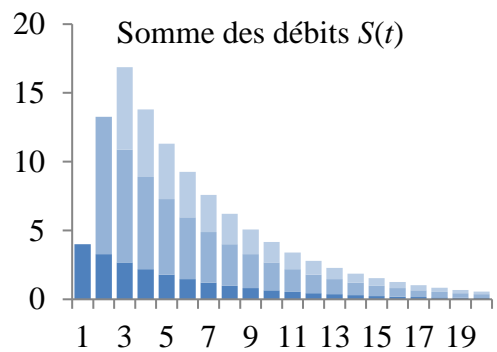
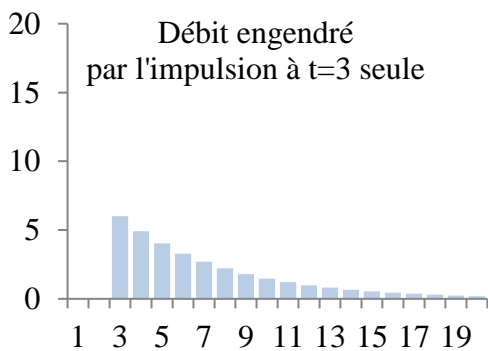
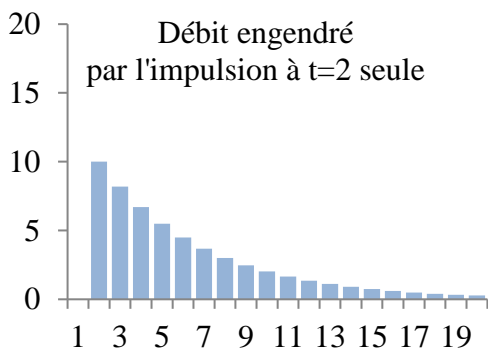
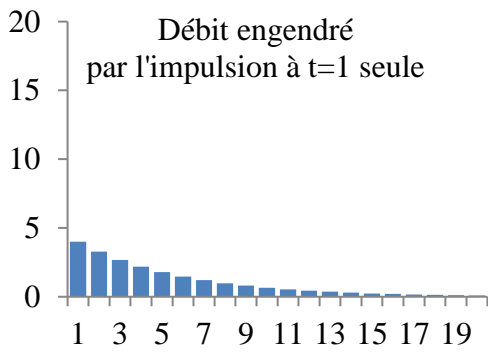
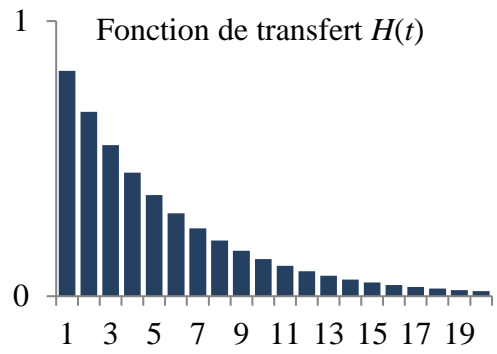
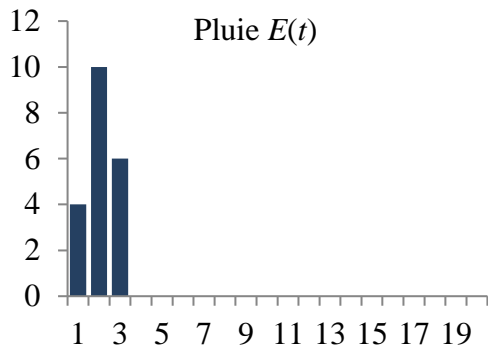


Figure I-7 : Trois impulsions (pluie), fonction de transfert, trois débits engendrés par les impulsions et résultat du produit de convolution (somme des débits)

En hydrologie, l'approche de modélisation par fonction de transfert découle du concept de l'hydrogramme unitaire apparu dans les années 1930 [Sherman, 1932; Horton, 1933].

L'hydrogramme (expression du débit en fonction du temps) unitaire d'un bassin est défini comme l'hydrogramme résultant d'une pluie nette de hauteur unité uniformément répartie sur tout le bassin. Si la durée de l'évènement pluvieux est suffisamment courte, il est désigné par le terme « hydrogramme unitaire instantané » (HUI), de la même manière qu'est définie la réponse impulsionnelle d'un système, vue précédemment. La linéarité du système est implicitement admise : le débit est directement proportionnel à la pluie (proportionnalité) et l'hydrogramme résultant d'une pluie de longue durée est égal à la somme des hydrogrammes unitaire élémentaire issus des pluies instantanées (additivité). Ce concept est utilisé pour la prévision des crues car l'HUI est caractéristique du bassin versant et permet de calculer le débit des crues futures si la pluie nette est connue [Bhunya et al., 2007; Singh et al., 2007]. Le suivi de l'hydrogramme se fait alors au pas de temps horaire car il faut une résolution temporelle assez fine pour suivre les crues.

Traditionnellement, les hydrogrammes unitaires étaient décomposés, souvent graphiquement, en plusieurs parties : le débit provenant du ruissellement (*quickflow*) et le débit de base (*baseflow*), provenant de la vidange de la nappe (Figure I-8). Dans cette conception, seul le ruissellement est responsable du pic de crues et l'écoulement issu des eaux souterraines ne fournit que le débit de base. Cette conception a été remise en question par des études utilisant des suivis chimiques [Pinder and Jones, 1969] et isotopiques [Weiler et al., 2003; Viville et al., 2006] démontrant que la partie du débit due aux écoulements provenant des eaux souterraines pouvait être plus importante que ce qui était admis jusque-là. La séparation de l'hydrogramme en débit rapide et de base est donc souvent effectuée de manière différente selon les auteurs (voir par exemple [Bonacci, 1993] et [Dewandel et al., 2003]), même si l'allure globale reste toujours la même.

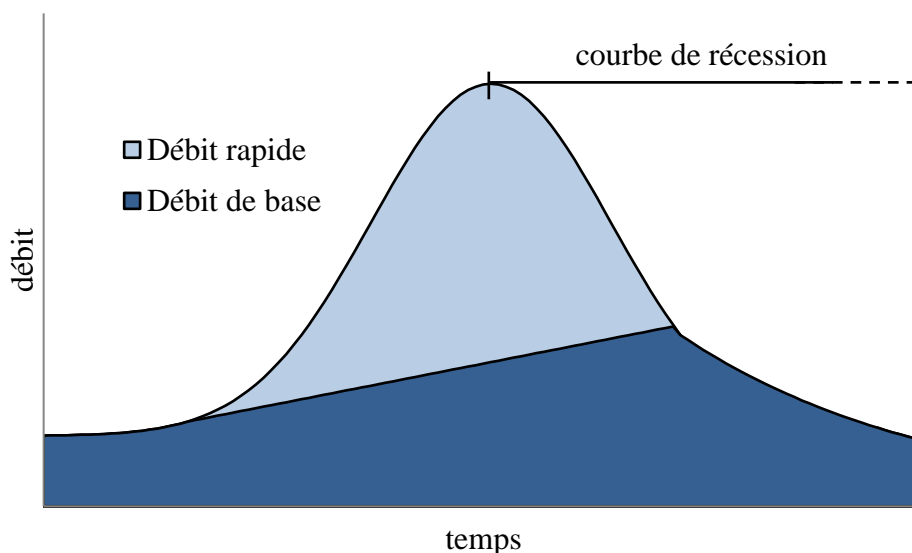


Figure I-8 : Composants de l'hydrogramme : débit rapide (*quickflow*) et débit de base (*baseflow*)

La partie décroissante de l'hydrographe unitaire, située après le débit de pic, est nommée « courbe de récession ». Cette courbe de récession présente généralement un point d'inflexion où la pente de la courbe change brusquement et à partir duquel le débit n'est censé provenir que des eaux souterraines. La partie de la courbe de récession située après ce point d'inflexion est appelée « courbe de récession du débit de base » (*baseflow recession curve*). L'étude de ces courbes de récession et du débit de base date de plus d'un siècle [Boussinesq, 1877; Maillet, 1905] et il existe donc un grand nombre de courbes modèles supposées approcher les hydrogrammes réels. Les équations des courbes issues de l'observation des phases de récession sont souvent utilisées en tant que fonction de transfert dans les modèles systémiques, particulièrement pour la modélisation des systèmes karstiques [Bonacci, 1993; Padilla et al., 1994].

Au vu des principes de modélisation par noyau de convolution, par hydrogramme unitaire ou par analyse des courbes de récession, il apparaît évident que l'élaboration des modèles systémiques repose finalement sur la recherche et l'identification de la fonction de transfert caractérisant le système. S'apercevant que la forme de l'hydrogramme unitaire (dans sa totalité ou seulement pour la partie « récession ») avait souvent une allure proche de certaines fonctions mathématiques, certains auteurs ont commencé à utiliser des fonctions paramétriques (FTP) en tant que fonction de transfert [Nash, 1957]. Comme leur nom l'indique, ces fonctions dépendent de plusieurs paramètres qui font varier leur forme (Figure I-9, a et b) :

$$H = f(t, \alpha, \beta, \dots, \zeta) \quad (\text{I.10})$$

où t représente le temps et $\alpha, \beta, \dots, \zeta$ représentent les différents paramètres de la FTP. Le problème consistant à caractériser un hydrosystème par une telle fonction de transfert se réduit alors à l'identification des paramètres de la FTP. Cette identification est aisée car le modèle possède peu de degrés de liberté, en l'occurrence les paramètres de la FTP dont le nombre est généralement compris entre 1 et 4. L'utilisation de ces fonctions de transfert est toujours d'actualité et l'identification de leurs paramètres est grandement facilitée par la puissance de calcul des ordinateurs modernes.

Les capacités de calcul ont également favorisé l'émergence des fonctions de transfert non-paramétriques (FTNP). Ces fonctions sont construites de manière discrète, valeur par valeur, par optimisation et peuvent être représentées par une somme de Dirac décalés dans le temps et pondérés :

$$H(t) = \sum_{i=1}^L K_i \cdot \delta_i(t) \quad (\text{I.11})$$

où L est la longueur de la fonction de transfert et K_i le poids du Dirac décalé de i . L'optimisation consiste à trouver les L paramètres $K_i \forall i \in [1, L]$ minimisant l'écart entre la sortie modélisée et la sortie mesurée. Comme les FTNP ne sont pas des lois mathématiques dépendantes d'un ou plusieurs paramètres mais une collection de valeurs scalaires pour des valeurs de temps relatifs donnés, leur forme n'est pas nécessairement régulière (Figure I-9, c).

Alors que les FTP peuvent se rapprocher des modèles conceptuels où une structure prédéfinie décrit le fonctionnement du système, les FTNP sont comparables aux réseaux de neurones car elles ne respectent aucun schéma déterminé et que seul compte l'ajustement de la simulation aux données. Dans la même veine, les réseaux de neurones ayant plus de degrés de liberté que les modèles conceptuels, les FTNP possèdent un nombre de degrés de liberté égal à leur longueur L , soit bien plus que les FTP.

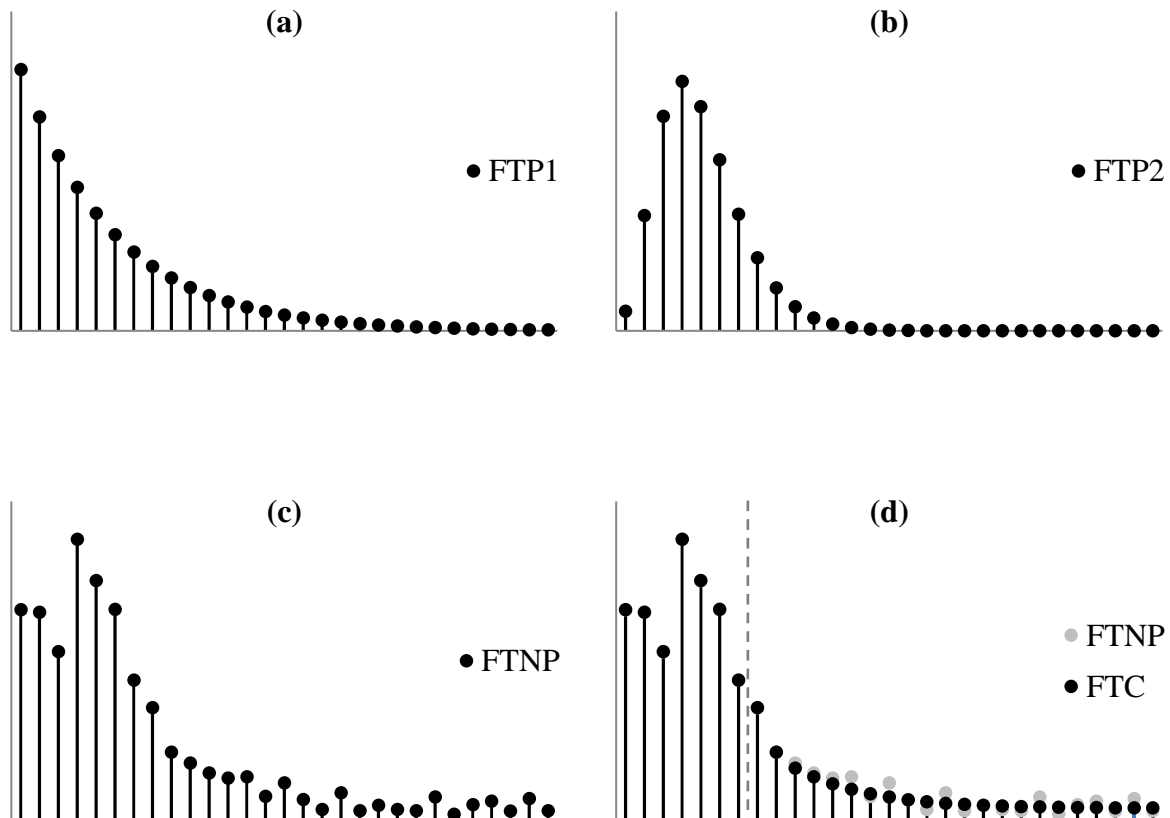


Figure I-9 : Exemple de fonctions de transfert : (a) Fonction de transfert paramétrique, loi Gamma ($\alpha=1$, $\beta=5$) ; (b) Fonction de transfert paramétrique, loi Gamma ($\alpha=5$, $\beta=1$) ; (c) Fonction de transfert non-paramétrique ; (d) Fonction de transfert composite

L'irrégularité des FTNP pose parfois des problèmes, en particulier pour leur partie temps longs où les oscillations rendent la fonction inutilisable pour la simulation des phases de récession de certains hydrogrammes, notamment ceux des sources karstiques. Certains auteurs [Denić-Jukić and Jukić, 2003] utilisent donc une fonction qu'ils nomment « fonction de transfert composite » (FTC) combinant une FTNP pour les temps courts afin d'avoir une bonne précision au niveau de l'intensité et du temps de pic, et une FTP pour les temps longs donnant des courbes de récession lisses pour mieux simuler les étiages des aquifères karstiques (Figure I-9, d).

I.5 Inversion du modèle

Une fois la structure du modèle hydrogéologique établie, le problème qui reste à résoudre est celui de l'inversion. Les principales connaissances disponibles sur l'hydrosystème sont généralement les données mesurées. En créant un modèle ajustable au moyen de paramètres on essaye de retrouver les données mesurées. Ce que l'on appellera le problème direct peut être représenté de la manière suivante :

$$\mathbf{y} = \mathbf{A}(\mathbf{p}) \quad (\text{I.12})$$

On suppose que l'opérateur direct $\mathbf{A} : \mathbb{R}^p \rightarrow \mathbb{R}^n$ est une application qui transforme un jeu de paramètre $\mathbf{p} \in \mathbb{R}^p$ en données observées $\mathbf{y} \in \mathbb{R}^n$. Dans le cas d'un problème linéaire discret et « bien posé » (i.e. le nombre de paramètres est égal au nombre de données observées), le problème devient trivial :

$$\mathbf{y} = \mathbf{A} \cdot \mathbf{p} \quad (\text{I.13})$$

où \mathbf{y} et \mathbf{p} sont deux vecteurs et \mathbf{A} la matrice d'observation. Sous couvert que la matrice \mathbf{A} soit inversible, le problème inverse se résume à chercher \mathbf{A}^{-1} tel que :

$$\mathbf{p} = \mathbf{A}^{-1} \cdot \mathbf{y} \quad (\text{I.14})$$

Cependant les problèmes d'hydrogéologie sont très fréquemment des problèmes dit « mal posés » : le nombre de données est généralement très différent du nombre de paramètres. Il est en général bien supérieur dans le cas d'approches conceptuelles et systémiques, bien inférieur dans le cas d'approches physiques distribuées. De plus, les données étant bruitées et entachées d'erreurs (de mesure, d'observation ou d'échantillonnage), la complexité des phénomènes et l'hétérogénéité du système dépassant de loin celles du modèle, il n'existe pas de solution qui permettrait de retrouver de manière exacte les données mesurées. L'approche classique du problème inverse consiste donc à chercher mathématiquement le jeu de paramètres qui minimise une fonction objectif, généralement une mesure de l'erreur entre la sortie mesurée et la sortie modélisée. La méthode dite « des moindres carrés » ayant comme mesure de l'erreur la somme du carré des résidus (résidu = valeur locale mesurée – valeur locale simulée) est utilisée dans de très nombreuses applications. Dans des cas dits "linéaires", une fonction objectif de type moindres carrés s'écrirait alors :

$$\Phi(\mathbf{p}) = (\mathbf{y} - \hat{\mathbf{y}})^T (\mathbf{y} - \hat{\mathbf{y}}) = (\mathbf{p}^T \mathbf{A}^T - \hat{\mathbf{y}}^T) (\mathbf{A} \mathbf{p} - \hat{\mathbf{y}}) \quad (\text{I.15})$$

avec $\hat{\mathbf{y}}$ le vecteur des observations, \mathbf{A} la matrice des relations linéaires entre modèle et paramètres, \mathbf{p} le vecteur des paramètres et \mathbf{y} le vecteur des variables de sortie du modèle. Minimiser $\Phi(\mathbf{p})$ revient à annuler ses dérivées partielles par rapport au vecteur \mathbf{p} . On obtient facilement le système d'équations identifiant des paramètres sous la forme :

$$\mathbf{A}^T \mathbf{A} \mathbf{p} = \mathbf{A}^T \hat{\mathbf{y}} \quad (\text{I.16})$$

Inverser des problèmes hydrologiques suppose néanmoins d'avoir recours à des algorithmes de résolution non-linéaire (car les données observées ne varient pas comme une combinaison linéaire des valeurs des paramètres [Davis, 1993]), comme les algorithmes à directions de descente et les métaheuristiques. Les algorithmes à directions de descente, comme les algorithmes de Gauss-Newton, du gradient (« *steepest descent* ») ou de Levenberg-Marquardt, cherchent à minimiser une fonction objectif de type moindres carrés en utilisant son gradient. Ces algorithmes sont expliqués plus en détail dans de nombreux travaux dont une synthèse très complète sur l'estimation de paramètres proposée par [Tarantola, 2005].

Le terme « métaheuristique » renvoie quant à lui à des algorithmes stochastiques itératifs utilisés indépendamment du problème considéré. Cette définition est assez large pour englober un grand nombre de techniques. La majeure partie de ces algorithmes est basée sur l'observation de comportements naturels : optimisation par essaim de particules, algorithme de colonies de fourmis, algorithmes évolutionnaires, recuit simulé, etc. L'avantage principal des métaheuristiques par rapport aux méthodes de résolution « classiques » est que la fonction objectif n'a pas besoin d'être dérivable par rapport aux paramètres du modèle. Incidemment, un calcul précis avec les méthodes de gradients suppose que l'on puisse identifier les sensibilités du modèle aux paramètres ou être capable de dériver les équations du problème direct par rapport aux paramètres. Les optimisations basées sur les méthodes de gradients sont donc efficaces si on les implante soit de manière intrusive, soit en calculant par perturbation les sensibilités aux paramètres. En d'autres termes, il faut soit modifier le code résolvant le problème direct pour accéder aux diverses expressions dont le moteur d'optimisation a besoin, soit s'assurer que les fonctions qui composent le modèle sont continûment dérivables. Les optimisations métaheuristiques se contentent d'observer les résultats fournis par le code qui résout le problème direct puis proposent par conjecture des modifications à apporter aux paramètres du modèle, sans condition sur la dérivabilité des fonctions du modèle. L'implémentation est non intrusive et par conséquent largement facilitée.

Avant de sélectionner une de ces métaheuristiques pour optimiser les paramètres d'un modèle il faut garder à l'esprit l'existence du théorème « No Free Lunch » [Wolpert and Macready, 1997]. Ce dernier stipule qu'il n'existe pas d'algorithme universellement plus efficace que tous les autres sur toutes les classes de problèmes existants. Pour autant, sur un problème donné, un algorithme peut être plus efficace qu'un autre, mais il ne le sera pas pour toutes les classes de problèmes. Il est donc important de choisir une métaheuristique qui soit efficace pour le problème considéré.

L'algorithme utilisé dans la suite de ces travaux (décrit plus en détail dans le chapitre suivant) est basé sur l'intelligence en essaim (*swarm intelligence*). Ce type d'algorithme utilise des populations d'individus, chacun à la recherche de la meilleure solution. Les règles suivies par chaque individu sont simples mais l'organisation d'une population permet de faire émerger certains comportements qualifiés d'intelligents. La métaheuristique utilisée dans le modèle présenté ici est de type « optimisation par essaim de particules » ou « *particle swarm* »

optimization » (PSO) et sera présentée plus en détails dans le chapitre III. Cet algorithme donne de bons résultats pour les problèmes dont la fonction objectif possède une topologie difficile pour l'optimisation, comme des minimums situés dans des vallées « plates » de gradients quasi-nuls [Fernández-Martínez et al., 2010]. De plus, il se révèle plus efficace en termes de convergence et de rapidité que les algorithmes génétiques, une autre métaheuristique très utilisée en modélisation hydrologique. [Kennedy and Spears, 1998] ont en effet montré dans une étude expérimentale utilisant un générateur de problème multimodaux que l'algorithme PSO atteignait l'optimum global plus rapidement que trois différentes versions d'algorithmes génétiques. De nombreux chercheurs utilisent donc désormais des algorithmes de type PSO pour optimiser les paramètres de leur modèle hydrologique [Chau, 2006; Tada, 2007; Jiang et al., 2010; Liu et al., 2013].

Sachant qu'il est impossible d'obtenir la valeur de la fonction objectif en tout point de l'espace des paramètres (donc de tester l'efficacité de tous les jeux possibles de paramètres) en un temps fini, les métaheuristicques peuvent apporter une réponse satisfaisante aux problèmes inverses en fournissant une solution acceptable, en un temps raisonnable. En revanche, il est généralement impossible de prouver que l'algorithme trouvera la ou les meilleures solutions. Le but est donc d'avoir un algorithme qui fonctionne pour la majeure partie des exemples et qui soit capable de fournir des solutions de bonne qualité dont on espère qu'elles seront proches de la solution optimale [Yang, 2012]. Étant donné que ces algorithmes doivent fonctionner pour un grand nombre de problèmes, il est nécessaire de trouver un équilibre entre leur capacité à résoudre des problèmes très spécifiques (en utilisant des informations sur le fonctionnement *a priori* du système) et des problèmes de type boîte noire (où le fonctionnement du système est inconnu). Ce compromis à faire entre exploration et exploitation est au cœur du développement des métaheuristicques [Blum and Roli, 2003]. Spécifiquement pour les algorithmes PSO, l'exploration désigne la capacité des particules à parcourir l'espace des paramètres, afin de générer une multitude de solutions éloignées, dans l'espoir de trouver un minimum qui soit optimal. L'exploitation désigne la capacité de tout ou partie des particules à converger rapidement vers un endroit correspondant à une bonne solution, ce qui va jouer sur la rapidité de la convergence de l'algorithme et donc le temps de calcul.

I.6 Complexité versus simplicité, biais versus variance

L'équilibre à trouver entre convergence et recherche trouve son parallèle lorsqu'on s'intéresse à la complexité des modèles, en considérant leur nombre de degrés de liberté (de paramètres). Intuitivement, on pourrait penser qu'un modèle inversé produira des simulations d'autant meilleures que le nombre de degrés de liberté est élevé. L'analogie mécanique est effectivement vraie : plus un système mécanique possède de degrés de liberté et plus les mouvements qu'il effectue sont variés. Mais cela est partiellement faux en modélisation, à cause justement des procédures d'inversion. [Denić-Jukić and Jukić, 2003] comparent les performances d'une FTC et d'une FTNP. La FTNP a un nombre de degré de liberté égal au nombre de pas de temps (soit dans l'exemple pris par les auteurs 200 paramètres). La FTC, composée d'une FTNP pour les temps courts (jusqu'à 17 ou 51 jours selon les modèles) et d'une FTP possédant entre 2 et 3 paramètres, dépend donc de 19 à 54 paramètres. Étant donné qu'il est théoriquement possible

pour la FTNP de prendre n'importe quelle forme, y compris la forme de la FTC, la FTNP devrait être celle qui trouve l'optimum de la fonction objectif. Or pour la donnée hydrologique sélectionnée, toutes les FTC ont des meilleures performances que la FTNP, aussi en bien en calibration qu'en prédiction. Cela peut s'expliquer par ce que [Bellman, 1961] nomme la « malédiction de la dimension » (« *curse of dimensionality* ») : 100 points dans l'intervalle unidimensionnel $[0,1] \in \mathbb{R}$ suffisent pour faire une analyse statistique de ces observations, alors que 100 points dans l'espace à 10 dimensions $[0,1]^{10} \in \mathbb{R}^{10}$ ne donnent pas une couverture suffisante pour réaliser une telle analyse. De la même manière, à chaque fois que le nombre de degrés de liberté d'un modèle croît d'une unité, l'espace des paramètres dans lequel les métaheuristiques cherchent un optimum augmente d'une dimension, rendant chacune des simulations de plus en plus isolées. Ainsi, une complexité excessive, c'est-à-dire une sur-paramétrisation peut s'avérer néfaste, raison pour laquelle les modèles de type conceptuel sont si prisés pour leur parcimonie. Il est donc nécessaire de trouver un équilibre entre un modèle complexe, capable de simuler un grand nombre de processus à l'aide de nombreux paramètres mais difficile à optimiser, et un modèle simple, dont l'optimisation est aisée mais qui peut s'avérer insuffisant au vu de la complexité des relations entre les entrées et les sorties. C'est ce que le modèle développé dans les travaux qui suivent tente de réaliser : un compromis entre facilité d'optimisation et description optimale des processus.

Cet équilibre nécessaire entre complexité et simplicité du modèle trouve son origine dans ce que l'on nomme en apprentissage automatique le dilemme biais-variance [Johannet, 2011]. En prenant en compte le bruit présent dans les observations, l'équation (I.12) peut s'écrire sous la forme

$$\mathbf{y} = \mathbf{A}(\mathbf{p}) + \mathbf{B} \quad (\text{I.17})$$

où \mathbf{B} est une variable aléatoire d'espérance mathématique nulle et de variance σ^2 représentant le bruit. Le but des modèles est de trouver la fonction $\hat{\mathbf{A}}(\mathbf{p})$ qui estime au mieux la fonction $\mathbf{A}(\mathbf{p})$ en fonction des données disponibles. Il est possible de démontrer que l'erreur de prédiction réalisée par le modèle est alors l'espérance $E\left[\left(\mathbf{A}(\mathbf{p}) + \mathbf{B} - \hat{\mathbf{A}}(\mathbf{p})\right)^2\right]$ qui peut se décomposer sous la forme :

$$E\left[\left(\mathbf{A}(\mathbf{p}) + \mathbf{B} - \hat{\mathbf{A}}(\mathbf{p})\right)^2\right] = \sigma^2 + \left(E\left[\hat{\mathbf{A}}(\mathbf{p})\right] - \mathbf{A}(\mathbf{p})\right)^2 + E\left[\left(\hat{\mathbf{A}}(\mathbf{p}) - E\left[\hat{\mathbf{A}}(\mathbf{p})\right]\right)^2\right] \quad (\text{I.18})$$

Le premier terme correspond au bruit irréductible présent dans les observations, le second terme est appelé « biais » et représente l'erreur due aux hypothèses erronées de l'algorithme d'apprentissage, et le troisième terme, appelé « variance », mesure l'erreur provenant de la sensibilité à de faibles fluctuations des données. Variance et biais variant en sens inverse, il est nécessaire de trouver un compromis quant à la complexité du modèle et au nombre de données d'entrée. Un biais élevé signifie que le modèle peine à trouver les relations adéquates entre les entrées et les sorties, le modèle est alors sous-ajusté. Au contraire, un modèle où la variance est

élevée aura tendance à être sur-ajusté en étant trop sensible aux spécificités de l'ensemble d'apprentissage, voire du bruit aléatoire présent dans celui-ci. Vu le nombre conséquent de données que le modèle présenté ci-après reçoit, par rapport au nombre limité de paramètres, le sous-ajustement n'est pas à craindre. Cependant, pour lutter contre le sur-ajustement, une méthode de régularisation passive (présentée au Chapitre IV :) sera mise en place : la validation croisée.

CHAPITRE II : DESCRIPTION DU MODÈLE MASH

Les modèles complexes à base physique mettent à profit l'expertise du modélisateur, ce que les modèles plus simples, à base conceptuelle ou systémique, ne permettent peu voire pas du tout. Toutefois, les approches simplifiées, grâce à leur rapidité et leur stabilité peuvent bénéficier d'outils externes sophistiqués d'estimation des paramètres ou d'analyse de sensibilité, auxquels les modèles complexes ne peuvent accéder [Doherty and Christensen, 2011]. Le modèle développé dans les travaux présenté ici se nomme MASH (« *Modélisation par Approche Systémique des Hydrosystèmes* ») et se situe à l'interface entre l'approche conceptuelle et l'approche systémique. De l'approche conceptuelle, il conserve la notion de pseudo-réservoirs interconnectés, agencés selon une architecture série et parallèle. De l'approche systémique, il garde la capacité de pouvoir s'adapter à une variété de problèmes et surtout d'utiliser des fonctions de transfert. Les deux principaux apports de ce modèle par rapport aux différents modèles existant sont : 1- son architecture versatile associant dans le même noyau de convolution d'une à douze fonctions de transfert paramétrique(s) choisie(s) dans une bibliothèque et, 2- son inversion réalisée par un algorithme performant issu de l'intelligence artificielle « en essaim ». Ces spécificités ainsi qu'une description exhaustive de MASH sont exposées ci-dessous.

II.1 Non-linéarité dans les modèles systémiques

La modélisation par fonction de transfert se base implicitement sur de nombreuses hypothèses quant au système considéré.

Le système doit tout d'abord être stationnaire, c'est-à-dire invariant dans le temps [Denić-Jukić and Jukić, 2003]. En effet, si la dynamique générale du système change au cours du temps, le noyau de fonctions de transfert caractérisant le système changera également. Dans ces conditions, un noyau invariant ne peut reproduire correctement toutes les relations entrée-sortie sur une période de temps déterminée. Il reste possible de traiter de systèmes non-stationnaires sous couvert de connaître l'enchaînement et la durée des périodes successives pour lesquelles on admet que le système est pseudo-invariant (les dénommées « *stress periods* » des modèles pseudo-instationnaires). Néanmoins, un système non stationnaire restera de comportement peu prédictible s'il est vu par une approche systémique ce qui réduit d'autant l'intérêt du modèle. De ce fait, les modèles basés sur les fonctions de transfert peuvent avoir des difficultés à simuler des bassins versant dont l'occupation des sols change durant les chroniques étudiées. En revanche, lorsque les entrées ou les sorties sont modifiées mais que cette variation est connue et mesurée, et que le système ne change pas, les modèles systémiques disposent de sérieux atouts.

Le système est considéré comme étant dynamique, dans le sens utilisé par les physiciens, c'est-à-dire qu'il est causal et déterministe. La notion de causalité, stipulant que l'avenir du système ne dépend que du passé et du présent, ne pose pas de problème et reste implicitement acceptée dans une large majorité de modèles. Le déterminisme, par contre, impose de considérer le système comme non-bruité, alors que le bruit est bien présent, qu'il provienne des systèmes de mesures ou de la complexité du système. Les données mesurées ne correspondent pas à la valeur

exacte des variables des hydrosystèmes, mais plutôt à une version bruitée et distordue de celles-ci. La qualité des données d'entrée-sortie (ou la capacité à filtrer le bruit) est donc un des points cruciaux de la modélisation systémique.

La dernière hypothèse forte émise pour les modèles utilisant le produit de convolution est la linéarité. La linéarité est la réunion de deux propriétés : la proportionnalité (ou homogénéité) et le principe de superposition (ou additivité). Pour un modèle hydrologique, la proportionnalité signifie que si par exemple l'entrée est multipliée par deux, la sortie l'est aussi, et le principe de superposition signifie que si on connaît la solution à deux problèmes élémentaires différents, chacun caractérisé par une série de contraintes sur le système, la solution du problème joint qui associe les contraintes des deux problèmes élémentaires est la somme des solutions élémentaires. Mathématiquement, une application f de E dans F , où E et F sont deux espaces vectoriels à gauche sur un corps K , est linéaire si et seulement si

$$\forall (x, y) \in E^2, \forall \mu \in K, f(x + \mu y) = f(x) + \mu f(y) \quad (\text{II.1})$$

Or les systèmes réels ne sont généralement pas linéaires. Le problème qui se pose généralement pour les modèles pluie-débit est que la proportionnalité n'est pas respectée : certains phénomènes n'apparaissent en effet qu'à partir d'un seuil de précipitation et par conséquent

$$f(\mu y) \neq \mu f(y) \quad (\text{II.2})$$

Cela peut poser des problèmes lors de simulations d'événements extrêmes, une application courante des modèles systémiques. Dans un modèle pluie-débit cherchant à reproduire les crues, le fait que le modèle soit linéaire entraîne une estimation erronée des événements extrêmes. Plusieurs moyens d'introduire des non-linéarités dans les modèles basés sur les fonctions de transfert ont donc été développés tels que l'utilisation de noyaux de Volterra, l'introduction d'une fonction de production ou l'ajout d'un seuil lors du calcul des entrées.

II.1.1 Noyaux de Volterra

Les aquifères karstiques constituent un exemple classique de systèmes complexes. Cette complexité vient de l'hétérogénéité structurelle du milieu, très contrastée et non corrélée, et des changements de physique associés à cette hétérogénéité (e.g. écoulement diffusif versus hyperbolique entre matrice et conduits). Leur organisation est souvent décrite à l'aide de trois parties : zone non-karstique, karst non-saturé et karst saturé [Mangin, 1975]. Afin de réussir à modéliser ces aquifères dont le comportement est non-linéaire, certains auteurs [Labat et al., 1999; Jukić and Denić-Jukić, 2006] utilisent comme opérateur intégral des noyaux dits « de Volterra ». Ceux-ci peuvent être vus comme une généralisation de l'opérateur de convolution :

$$S(t) = \sum_n \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} H_n(\tau_1, \dots, \tau_n) E(t - \tau_1) \dots E(t - \tau_n) d\tau_1 \dots d\tau_n \quad (\text{II.3})$$

où $H_n(\tau_1, \dots, \tau_n)$ représente les noyaux de Volterra. Cette formule est utilisée jusqu'à $n=2$. Pour $n=1$, on retrouve bien le produit de convolution de deux fonctions, et pour $n=2$ on obtient le noyau non-linéaire :

$$S(t) = \int_{-\infty}^{+\infty} H_1(\tau)E(t-\tau)d\tau + \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} H_2(\tau_1, \tau_2)E(t-\tau_1)E(t-\tau_2)d\tau_1d\tau_2 \quad (\text{II.4})$$

Notons que la non linéarité est ici matérialisée par le fait que la sortie puisse être fonction du signal d'entrée élevé au carré. Dans leur étude, [Labat *et al.*, 1999] montrent que le noyau non-linéaire d'ordre 2 est plus performant que le noyau linéaire d'ordre 1 pour des chroniques de 30 ans au pas journalier, mais par pour des chroniques de 1 an au pas de temps semi-horaire. [Jukić and Denić-Jukić, 2006] évoquent pour leur part la difficulté d'utilisation de ces noyaux pour effectuer des prédictions ainsi que leur manque de sens physique. Pour ces raisons, ce genre de noyaux n'est pas utilisé dans MASH.

II.1.2 Fonction de production

La manière la plus simple d'introduire une non-linéarité dans les modèles conceptuels et systémiques est l'utilisation d'une fonction de production en amont de la fonction de transfert. Celle-ci est chargée de transformer les chroniques d'entrées brutes en chroniques "filtrées" qui intègrent les non-linéarités et autres effets de seuils. On prendra pour exemple celui bien connu de la transformation des chroniques de précipitation « brutes » (issues directement des mesures) en chroniques de précipitation « efficaces » ou « utiles » (qui représentent la portion des précipitations contribuant à la recharge des réserves en eau du sol). Cette fonction de production qui modélise les processus d'infiltration s'exprime par de nombreuses fonctions mathématiques : formule de Horton, de Dooge, de Green et Ampt, etc.

Dans MASH, la fonction de production agit comme un réservoir à trop plein, caractérisé par une réserve utile maximum R_{\max} (Figure II-1). Elle ne fait que représenter la manière très conventionnelle de calculer une pluie efficace. La première étape consiste à calculer l'état de la réserve utile R (le remplissage du réservoir) pendant l'intervalle i . Si la pluie est supérieure à l'évapotranspiration ($P_i \geq ETP_i$) :

$$\begin{cases} R_i = R_{\max} & \text{si } R_{i-1} + (P_i - ETP_i) \geq R_{\max} \\ R_i = R_{i-1} + (P_i - ETP_i) & \text{sinon} \end{cases} \quad (\text{II.5})$$

La réserve utile se remplit par conséquent de la différence entre la pluie et l'ETP si cette différence est positive, jusqu'à atteindre R_{\max} . Si la pluie est inférieure à l'évapotranspiration ($P_i < ETP_i$) :

$$\begin{cases} R_i = 0 & \text{si } R_{i-1} + (P_i - ETP_i) \leq 0 \\ R_i = R_{i-1} + (P_i - ETP_i) & \text{sinon} \end{cases} \quad (\text{II.6})$$

La réserve utile se vide par conséquent de la différence entre la pluie et l'ETP si cette différence est négative, jusqu'à atteindre 0. Finalement, la pluie efficace P_{eff} est égale à ce qui déborde du réservoir quand celui-ci est plein, c'est-à-dire quand $R_i = R_{max}$:

$$\begin{cases} P_{eff} = (P_i - ETP_i) - (R_{max} - R_{i-1}) & \text{si } R_i = R_{max} \\ P_{eff} = 0 & \text{sinon} \end{cases} \quad (\text{II.7})$$

Cette fonction de production est non-linéaire car elle ne respecte pas le principe de proportionnalité : la pluie efficace n'est pas proportionnelle à la pluie brute.

Les fonctions de production sont nécessaires pour modéliser certains comportements comme la modification du fonctionnement de l'hydrosystème pendant les différentes saisons. Pendant toute une partie de l'année (généralement d'avril à septembre pour les climats tempérés) la réserve utile ne dépasse jamais la réserve utile maximum, et les précipitations ne contribuent jamais à la recharge des aquifères. Pendant le reste de l'année, les précipitations sont suffisamment fortes, et surtout l'ETP est suffisamment faible pour permettre une recharge des réserves en eau.

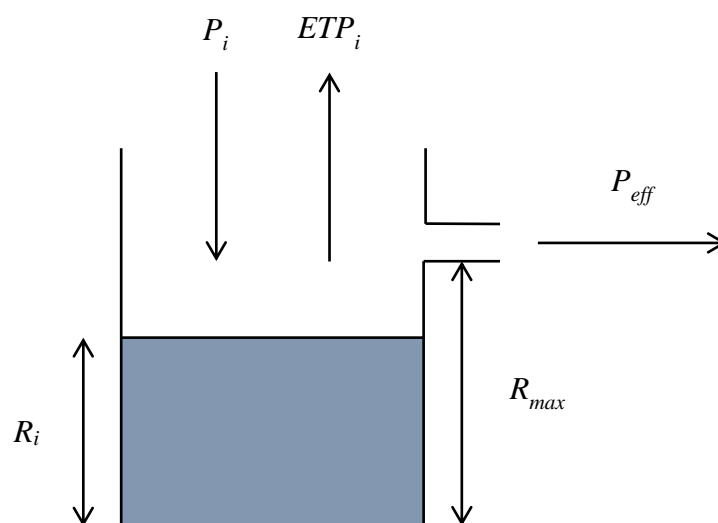


Figure II-1: Schéma de fonctionnement de la fonction de production

II.1.3 Seuillage des entrées

Une autre méthode utilisée dans MASH pour introduire une non-linéarité dans le modèle est le seuillage optionnel de certaines des entrées. En effet, on suppose ici que pour des valeurs de précipitations efficaces supérieures à un certain seuil, de nouveaux phénomènes apparaissent, se rajoutant aux phénomènes précédents. Par exemple, lors de certains épisodes orageux sur des sols cultivés de manière intensive ou présentant des horizons argileux peu profonds, un écoulement hypodermique peut se mettre en place, en plus des écoulements usuels (ruissellement, vidange de nappe, etc.). Pour prendre en compte ces écoulements de subsurface

dont le temps de transfert moyen est plus long que l'écoulement superficiel, mais plus court que l'écoulement différé des nappes profondes, il faut ajouter une branche au modèle qui utilise en entrée non pas la chronique de pluie efficace mais une chronique de pluie efficace seuillée. Cette dernière est calculée très simplement, en fonction de la valeur de pluie seuil P_{\min} :

$$P_{seuil} = \begin{cases} 0 & \text{si } P_{eff} < P_{limite} \\ P_{eff} & \text{sinon} \end{cases} \quad (\text{II.8})$$

Cette modification des entrées est aussi non-linéaire à cause du non-respect du principe de proportionnalité.

Les deux modifications possibles avec MASH (fonction de production et seuillage des entrées) présentées ici ne portent pas sur la structure interne du modèle, qui reste composée de fonctions de transfert convoluées linéairement avec les entrées. Cette architecture s'oppose donc à celle des noyaux de Volterra, où la non-linéarité provient des fonctions de transfert qui sont au cœur du modèle. Toutefois, les possibilités offertes par la modification des entrées permettent d'utiliser plusieurs fois la même entrée sur des branches parallèles différentes. En effet, pour un modèle à deux branches parallèles, composées respectivement des fonctions de transfert H_1 et H_2 , si l'entrée E est la même dans les deux branches :

$$S(t) = E * H_1(t) + E * H_2(t) \quad (\text{II.9})$$

$$S(t) = E * (H_1 + H_2)(t) \quad (\text{II.10})$$

vu que le produit de convolution est distributif sur l'addition. L'ajout d'une autre branche est donc inutile car l'équation (II.10) montre qu'il existe une fonction de transfert équivalente égale à $H_1 + H_2$ privilégiant l'utilisation d'une branche unique. Si en revanche on utilise deux versions modifiées différemment de la même entrée, en jouant sur les paramètres de la fonction de production ou le seuillage, on a alors :

$$S(t) = E_1 * H_1(t) + E_2 * H_2(t) \quad (\text{II.11})$$

Ainsi, les deux branches sont utiles et indispensables à la modélisation des phénomènes non-linéaires. Cette structure acceptant différentes entrées, convoluées par différentes fonctions de transfert dans des branches parallèles est rendu possible par l'architecture particulière adoptée pour MASH.

II.2 Architecture et versatilité du modèle

Une des principales différences entre MASH et les autres modèles systémiques ou conceptuels réside dans son architecture particulière associant fonction de transfert en série et en parallèle, ainsi que dans sa versatilité lui permettant de s'adapter à un grand nombre de situation, étant

donné que le modèle peut utiliser d'une à douze fonctions de transfert (quatre branches en parallèle de chacune trois fonctions en série).

II.2.1 Versatilité du modèle

L'architecture de MASH est beaucoup plus libre que celles des modèles conceptuels. En effet, comme on l'a vu au chapitre précédent, ceux-ci sont composés d'un nombre prédéfini d'entrées et de réservoirs, limitant les possibilités d'application à une grande variété de problèmes. À l'opposé, la majorité des modèles systémiques font preuve d'une grande souplesse quant à leur capacité d'adaptation. Les modèles basés sur les réseaux de neurones peuvent ainsi simuler des phénomènes très variés, selon le nombre d'entrées, de neurones et de couches de neurones qu'ils possèdent. MASH a également été développé dans cette optique afin de disposer d'un outil de modélisation par fonctions de transfert adapté à de nombreuses applications. Cette adaptabilité se traduit en pratique par trois caractéristiques du modèle laissées à discrétion de l'utilisateur pour chaque nouveau problème : le nombre d'entrées, le nombre de fonctions de transfert en série et le type de fonctions de transfert utilisé.

Le nombre d'entrées, compris entre un et quatre, définit également le nombre de branches parallèles que comporte le modèle. Le nombre de fonctions de transfert en série, compris entre un et trois, impose la taille de chacune des branches. Le type de fonction de transfert paramétrique est choisi pour chacune des fonctions du modèle au sein d'une bibliothèque comportant actuellement quatre fonctions type mais pouvant être étendue ultérieurement.

MASH possède donc dans sa version la plus simple une entrée et une fonction de transfert (**Figure II-2**), et dans sa version la plus complexe quatre entrées et douze fonctions de transfert (**Figure II-3**).

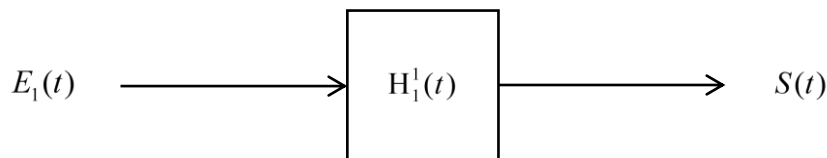


Figure II-2 : Architecture MASH la plus simple : une entrée et une fonction de transfert

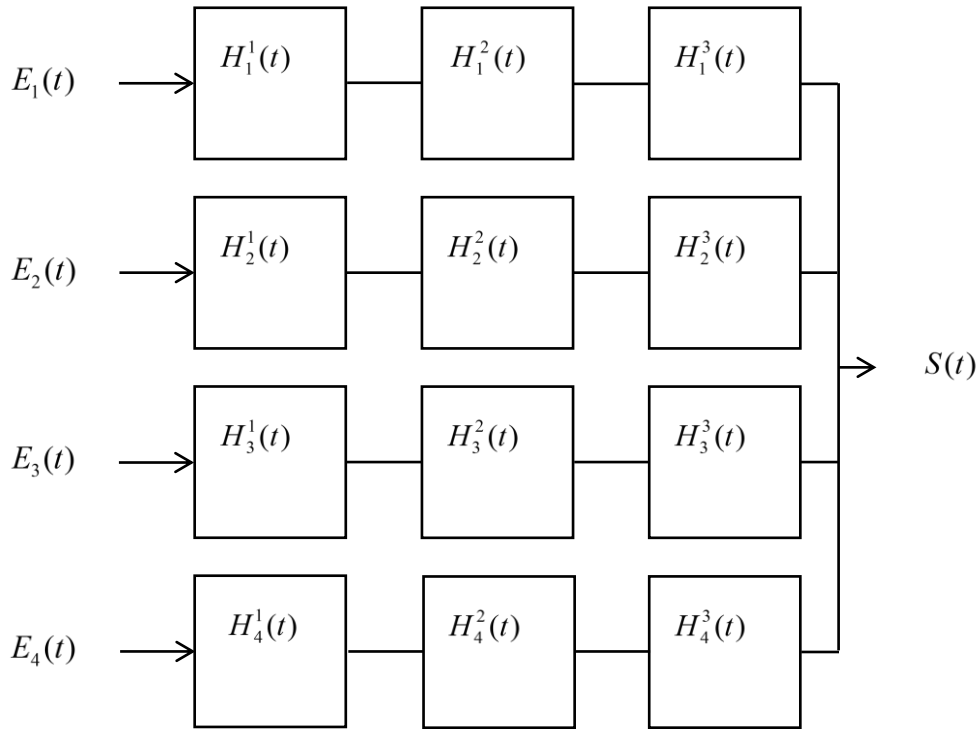


Figure II-3 : Architecture MASH la plus complexe : quatre entrées et douze fonctions de transfert

La sortie $S(t)$ est calculée comme étant la somme des produits de convolutions des entrées par les fonctions de transfert de chaque branche :

$$S(t) = E_1 * H_1^{eq}(t) + E_2 * H_2^{eq}(t) + E_3 * H_3^{eq}(t) + E_4 * H_4^{eq}(t) \quad (\text{II.12})$$

où E_i est la $i^{\text{ème}}$ entrée du modèle et H_i^{eq} est la $i^{\text{ème}}$ fonction de transfert équivalente égale au produit de convolution des trois fonctions de transfert élémentaires d'une branche :

$$H_i^{eq}(t) = K_i^{eq} \cdot (H_i^1 * H_i^2 * H_i^3(t)) \quad (\text{II.13})$$

où K_i^{eq} est le gain de la $i^{\text{ème}}$ branche et H_i^α est la $\alpha^{\text{ème}}$ fonction de transfert de la $i^{\text{ème}}$ branche. Les fonctions de transfert élémentaires inutilisées sont égales à la distribution de Dirac étant donné que celle-ci est l'élément neutre du produit de convolution.

En couplant cette architecture modulaire avec la possibilité de choisir individuellement chacune des fonctions élémentaires de transfert, les capacités d'adaptation du modèle sont importantes. Dans les modèles à base physique, alors même que les équations explicitant les transferts sont *a priori* complètement définies en amont au sein du logiciel, l'utilisateur crée un modèle sur mesure adapté au problème (en définissant l'extension du maillage, le nombre de couches géologiques aquifères et aquitards, etc.). Dans une moindre mesure, MASH est également un générateur de modèle à mesure, plus qu'un système figé. En dépit du postulat sur des relations

de convolution entre les entrées et la sortie, de nombreuses options sont laissées à l'appréciation du modélisateur afin qu'il dispose d'un modèle adapté au problème qu'il désire résoudre, plutôt que d'essayer de contraindre le problème afin que celui-ci s'adapte au modèle.

II.2.2 Architecture en série et en parallèle

Lorsque deux fonctions de transfert $H_s^1(t)$ et $H_s^2(t)$ sont associées en série (**Figure II-4**), la sortie $S(t)$ est égale à :

$$S(t) = H_s^1 * H_s^2 * E(t) \quad (\text{II.14})$$

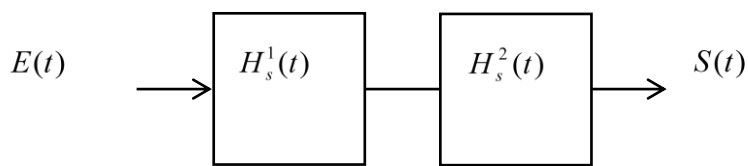


Figure II-4 : Deux fonctions de transfert en série

Lorsque deux fonctions de transfert $H_1^p(t)$ et $H_2^p(t)$ sont associées en parallèle (**Figure II-5**), la sortie $S(t)$ est égale à :

$$S(t) = H_1^p * E_1(t) + H_2^p * E_2(t) \quad (\text{II.15})$$

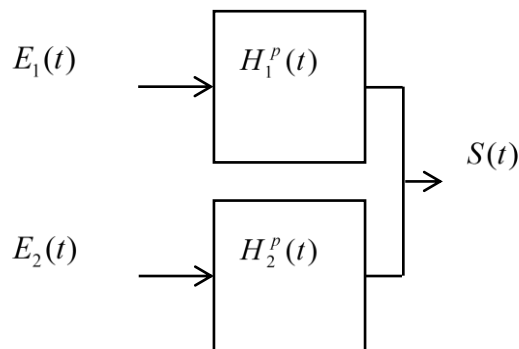
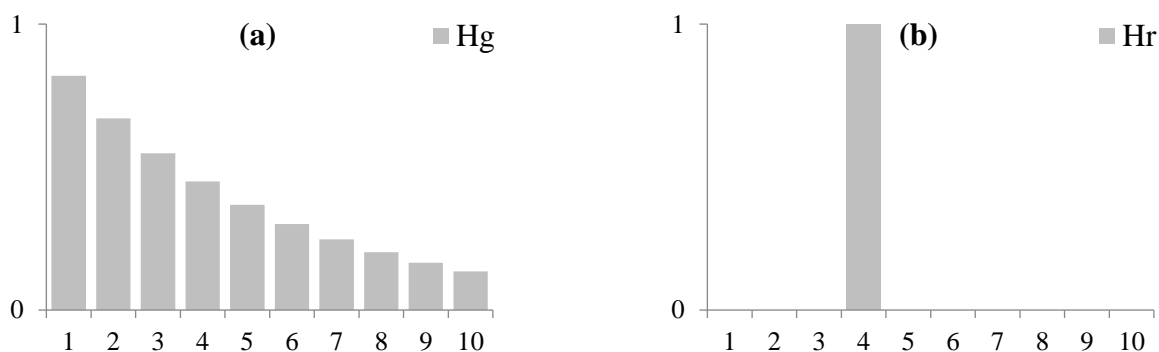


Figure II-5 : Deux fonctions de transfert en parallèle

L'utilisation de fonctions de transfert en série permet de modéliser des phénomènes ayant lieu successivement (**Figure II-6**, a, b et c). Ainsi, dans les modèles conceptuels, la fonction de production est en série avec la fonction de transfert : les chroniques de précipitations brutes sont d'abord transformées en chronique de précipitations efficaces par la fonction de production, puis en chronique de débit par la fonction de transfert, suivant ainsi le parcours réel de l'eau à

travers l'hydrosystème. De la même manière, les fonctions de transfert en série de MASH représentent les différents phénomènes ayant lieu les uns à la suite des autres. On peut par exemple imaginer d'utiliser trois fonctions de transfert en série : la première, une loi normale, traduira le caractère diffusif de l'infiltration à partir de la surface du bassin versant ; la deuxième, une loi Gamma, simulera la vidange de l'aquifère ; la troisième, une fonction retard pur, représentera le transfert entre matrice et conduit karstique.

L'utilisation de fonctions de transfert en parallèle permet de modéliser des phénomènes ayant lieu simultanément (**Figure II-6**, a, b et d). Ainsi, dans les modèles conceptuels, la fonction de transfert correspondant aux écoulements dits « rapides » est en parallèle avec la fonction de transfert correspondant aux écoulements dits « lents » : les chroniques de précipitations efficaces sont séparées en deux composantes qui forment les entrées des fonctions « rapide » et « lente ». Les fonctions de transfert en parallèle de MASH représentent différents phénomènes d'occurrence simultanée. Ces fonctions en parallèle peuvent par exemple être utilisées pour modéliser différents types d'écoulements : une fonction pour les écoulements à temps caractéristique court comme le ruissellement et une fonction pour les écoulements lents comme la vidange d'une nappe. La précipitation efficace tombée à un instant t participera à ces deux écoulements. La mise en parallèle des fonctions de transfert permet également de rendre compte d'une certaine hétérogénéité des hydrosystèmes qui ne va pas forcément de pair avec les modèles systémiques. On peut en effet considérer que les branches parallèles simulent divers sous-bassins ou des zones ayant des comportements (comme des temps de transfert moyens) différents. Les fonctions parallèles peuvent ainsi représenter des entités ayant des géologies et/ou des géomorphologies différentes. À noter toutefois qu'il est nécessaire d'utiliser des chroniques d'entrées différentes pour chacune des branches parallèles eût égard à la distributivité du produit de convolution sur l'addition. Il faudra utiliser des chroniques pluviométriques relevées à des stations différentes, ou jouer sur les paramètres de réserve utile maximum ou de seuil afin d'obtenir des entrées différenciées. Enfin, l'utilisation de branches parallèles permet d'utiliser des entrées de nature différente : il est par exemple possible d'utiliser une chronique de précipitations pour simuler la composante haute fréquence de la chronique de débit, et une chronique de hauteurs piézométriques pour en simuler la composante basse fréquence.



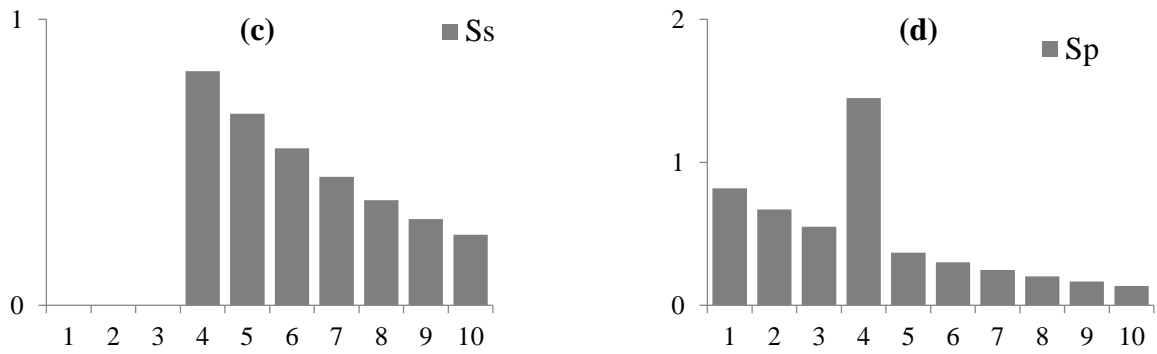


Figure II-6 : Architecture série et parallèle : (a) Fonction de transfert type loi Gamma H_g ; (b) Fonction de transfert type retard pur H_r ; (c) Sortie pour l'architecture série $S_s(t) = H_r * H_g * \delta(t)$; (d) Sortie pour l'architecture parallèle $S_p(t) = H_r * \delta(t) + H_g * \delta(t)$

II.2.3 Types de mise en relation possibles

Le choix des types de chroniques utilisés en entrée et en sortie est laissé à la libre appréciation du modélisateur. L'utilisation de MASH la plus classique est la construction d'un modèle « pluie-débit » : une chronique de précipitation est associée à une chronique d'ETP afin de calculer la pluie efficace qui sert d'entrée au modèle, et la sortie est une chronique de débit. Pour affiner certains cas de figure, il est possible d'ajouter dans une autre entrée comme une chronique de hauteur piézométrique (voir paragraphe précédent). D'autres types de modèles sont également possibles, comme les modèles « pluie-hauteur piézométrique » [Pinault and Schomburgk, 2006; von Asmuth et al., 2012]. Les données entrées sont alors des chroniques de précipitations et les données de sorties sont des chroniques de hauteurs piézométriques. Enfin, les modèles à fonctions de transfert permettent de suivre des concentrations d'éléments chimiques [Pinault et al., 2001] en utilisant en entrée des chroniques de précipitations et en sortie des chroniques de concentrations.

Dans MASH, il est également possible d'utiliser n'importe quel type de chronique en entrée comme en sortie : précipitation, hauteur piézométrique, concentration, température, etc. La seule contrainte est de s'assurer que le pas de temps de toutes les chroniques est le même. Cette restriction facilite le calcul des intégrales de convolution. Elle pourrait être levée en utilisant des interpolateurs de données manquantes ou de mise à l'échelle des chroniques. Cette option n'est pas implémentée pour l'instant sachant que toute interpolation est susceptible de produire un biais dans l'appréciation de la qualité des résultats. Deux interpolateurs utilisant le même support de pondération des données et le même calcul des poids engendreront nécessairement des corrélations (potentiellement) artificielles entre entrées et sortie. Comme pour les réseaux de neurones, l'algorithme d'optimisation a la possibilité de rendre la contribution d'une entrée quasi-nulle, via le paramètre de gain notamment. Il est donc possible de créer un modèle comportant plusieurs entrées testées séquentiellement afin de discriminer celles qui améliorent l'ajustement de la sortie. Au cours d'une inversion, si les paramètres de gain sont entrés comme paramètre d'ajustement, l'optimisation doit pouvoir automatiquement éliminer (gain faible) les branches inutiles. Cela étant, mettre des branches en parallèle dans un problème de convolution

c'est accroître le nombre de solutions optimales possibles. L'utilisateur devra par conséquent être vigilant et sélectionner sur des critères factuels les solutions qui semblent réalistes. On peut donc très bien, dans un modèle pluie-débit, ajouter une chronique de hauteurs piézométriques ou de températures, pour voir si ces dernières permettent de recréer un débit plus proche du débit mesuré que ne le permet un modèle n'utilisant qu'une chronique de précipitations. Si les branches additionnelles s'avèrent utiles à la modélisation, on pourra alors s'interroger sur la raison pour laquelle une entrée améliore le signal de sortie : la chronique de hauteurs piézométriques traduira peut-être les liens nappes-rivières, ou améliorera la simulation des périodes de récession, et la chronique de températures permettra peut-être d'incorporer une tendance à la modification des effets évapo-transpiratoires sur le système. Cette adaptation à une grande variété de chroniques d'entrées et de sortie est encore améliorée par la possibilité de choisir le type de fonction de transfert paramétrique utilisé pour chaque simulation et par l'ajout possible de fonction de transfert si les fonctions de la bibliothèque ne suffisent pas.

II.3 Fonctions de transfert

L'utilisation de fonctions de transfert dans les modèles systémiques prend sa source dans les notions d'hydrogramme unitaire et de courbe de récession. Étant donné que ces méthodes existent depuis plus d'un siècle, un nombre conséquent de fonctions de transfert existe, rendant par conséquent non-exhaustive l'énumération qui suit. Sont présentées dans ce paragraphe des fonctions de transferts issues de l'analyse des courbes de récession et d'autres provenant des probabilités. Les fonctions sélectionnées pour figurer dans la bibliothèque de MASH seront décrites plus en détail.

II.3.1 Courbes de récession

Les courbes de récession, utilisées à l'origine pour reproduire l'allure de la partie décroissante des hydrogrammes, sont fréquemment employées comme fonctions de transfert [Pinault *et al.*, 2001; Denić-Jukić and Jukić, 2003; Fleury *et al.*, 2007]. Ces courbes de récession, synonymes de variations de débit au cours du temps $Q = f(t)$, sont utilisées telles quelles en tant que fonction de transfert i.e. $H = f(t)$ dans des modèles pluie-débit où $Q(t) = H * P(t)$, avec $P(t)$ la chronique des pluies plus ou moins retravaillée. Des articles recensent de nombreuses fonctions décrivant les courbes de récession, fournissant ainsi de nombreuses FTP [Tallaksen, 1995; Griffiths and Clausen, 1997; Moore, 1997; Dewandel *et al.*, 2003]. Ces fonctions peuvent être empiriques, conceptuelles ou à base physique, selon une classification similaire à celle des différents types de modèle présentés dans le chapitre précédent.

Les deux formules « historiques » de l'analyse des courbes de récession sont celles de Maillet [Maillet, 1905] et de Boussinesq [Boussinesq, 1904]. La formule de Maillet est une équation exponentielle, solution analytique approchée de l'équation de diffusion en milieu poreux mais solution exacte de la vidange d'une cuve à surface libre (Figure II-7) :

$$Q(t) = Q_0 e^{-t/k} \quad (\text{II.16})$$

où $Q(t)$ est le débit au temps t , Q_0 le débit initial, et k le coefficient de récession. Ces deux paramètres peuvent être reliés à la géométrie de l'aquifère [Boussinesq, 1877].

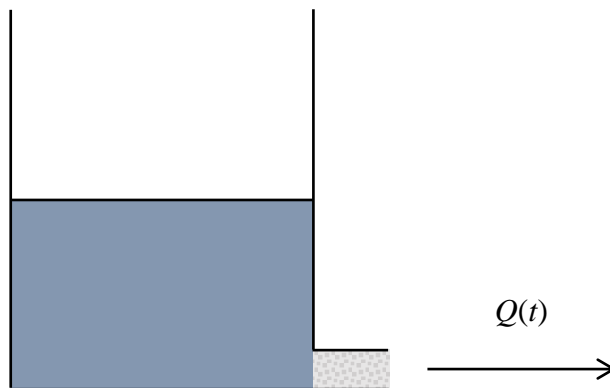


Figure II-7 : Schéma de fonctionnement du modèle de Maillet, d'après [Dewandel et al., 2003]

La formule de Boussinesq [Boussinesq, 1904] est une équation quadratique, solution exacte de l'équation de diffusion en milieu poreux. Sous certaines hypothèses (aquifère libre, homogène et isotrope, capillarité négligée, etc.), la solution de Boussinesq est une équation quadratique permettant d'exprimer le débit de vidange d'une nappe par la formule suivante :

$$Q(t) = \frac{Q_0}{(1 + \alpha t)^2} \quad (\text{II.17})$$

où Q_0 est le débit initial et α le coefficient de récession caractérisant l'aquifère. Le modèle de réservoir linéaire de Maillet simule bien la récession d'aquifères où la composante verticale des écoulements est importante, alors que le modèle de Boussinesq est plus adapté au cas où c'est la composante horizontale qui prédomine [Dewandel et al., 2003]. L'anisotropie de la perméabilité de l'aquifère change donc la forme de la courbe de récession.

Le modèle empirique dit « double exponentielle de Horton » apparaît pour la première fois dans [Horton, 1933]. Il est défini par ;

$$Q(t) = Q_0 e^{-t/k} \quad (\text{II.18})$$

où $Q(t)$ est le débit au temps t , Q_0 le débit initial, k le coefficient de récession et n une constante. Ce modèle se veut être une généralisation du réservoir linéaire $Q(t) = Q_0 e^{-t/k}$.

[Griffiths and Clausen, 1997] utilisent dans leur étude de nombreuses courbes de récession. Pour caractériser les écoulements issus d'eau stockée dans des dépressions en surface (lac, étang, marais, etc.), ils démontrent que la vidange d'un lac à travers un canal rectangulaire en pente raide peut s'écrire de la manière suivante :

$$Q(t) = \frac{a}{(1+bt)^3} \quad (\text{II.19})$$

où a et b sont des paramètres dépendants de la géométrie du lac et du canal.

En ce qui concerne les régions où les écoulements proviennent principalement de la fonte de neige et de glace, [Toebes and Strang, 1964] utilisent deux modèles empiriques:

$$Q(t) = a + \frac{b}{t^n} \quad (\text{II.20})$$

et
$$Q(t) = a + (Q_0 - a)e^{-\alpha t} \quad (\text{II.21})$$

où α est le coefficient de récession, et a, b et n sont des constantes. Dans ces deux formules, le débit tend asymptotiquement vers la constante a . [Kupczyk et al., 1994] interprètent cette constante comme le débit de base des rivières. Pour $a = 0$, l'équation (II.21) est équivalente à l'équation (II.16) de la formule de Maillet.

[Moore, 1997] fait le lien entre les équations exprimant le débit en fonction du stockage dans un réservoir et les fonctions de récession associées. Par exemple, le débit en fonction du stockage pour un réservoir linéaire est égal à :

$$Q = \frac{S}{k} \quad (\text{II.22})$$

avec Q le débit, S le stockage du réservoir (en unité de volume) et k le coefficient de récession. Ceci permet de retrouver la formule (II.16) en utilisant le principe de conservation. Selon des raisonnements analogues, il est possible de définir d'autres fonctions de transfert en utilisant des lois de stockage non-linéaires. [Beven and Kirkby, 1979] utilisent pour leur modèle TOPMODEL un réservoir exponentiel en prenant comme hypothèse une conductivité hydraulique du sol qui décroît exponentiellement en fonction de la profondeur :

$$Q = a e^{-b.DS} \quad (\text{II.23})$$

où a et b sont des constantes, et DS le déficit de stockage du bassin (en unité de volume). Ils arrivent ainsi à la fonction de récession :

$$Q(t) = \frac{Q_0}{1+bQ_0 t} \quad (\text{II.24})$$

De la même manière [Moore, 1997] à partir d'un réservoir suivant une loi de puissance :

$$Q = \alpha S^\beta \quad (\text{II.25})$$

définit la fonction :

$$Q(t) = Q_0(1 + \mu t)^p \quad (\text{II.26})$$

avec
$$p = \beta / (1 - \beta) \quad (\text{II.27})$$

et
$$\mu = \alpha^{1/\beta} (\beta - 1) Q_0^{(\beta-1)/\beta} \quad (\text{II.28})$$

où Q_0 est le débit initial, α et β des constantes. Cette équation, très générale, permet de retrouver de nombreux autres modèles :

- le modèle de Maillet (équation (II.16)) avec $\beta = 1$ dans l'équation (II.25)
- le modèle de Drogue (équation (II.41)) avec $p = -n$ et $\mu = \alpha$ (présentée au paragraphe suivant)
- le modèle de Boussinesq (équation (II.17)) avec $p = -2$ et $\mu = \alpha$
- le modèle de Griffiths (équation (II.19)) avec $p = -3$ et $\mu = b$

L'équation (II.26) est également équivalente à celle du modèle de [Coutagne, 1948] :

$$Q(t) = Q_0 [1 + (n-1)\alpha t]^{n/(1-n)} \quad (\text{II.29})$$

avec
$$\alpha = -\mu(1+p) \quad (\text{II.30})$$

et
$$n = \frac{p}{1+p} \quad (\text{II.31})$$

où α et n sont des constantes. De la même manière, l'équation (II.26) est équivalente à celle du modèle que l'on trouve dans [Wittenberg, 1999] sous la forme :

$$Q(t) = Q_0 \left[1 + \frac{(1-b)Q_0^{1-b}}{ab} t \right]^{\frac{1}{b-1}} \quad (\text{II.32})$$

avec
$$a = -\frac{1}{\mu(p+1)} Q_0^{-1/p} \quad (\text{II.33})$$

et
$$b = \frac{p+1}{p} \quad (\text{II.34})$$

où a et b sont des constantes.

Un autre exemple de loi de stockage, utilisée pour simuler le comportement de deux réservoirs indépendants en série ayant chacun leur propre cinétique de vidange :

$$Q = \alpha_2 S_2 \quad (\text{II.35})$$

et

$$\frac{dS_1}{dt} = Q_1 e^{-\alpha_1 t} - \alpha_2 S_2 \quad (\text{II.36})$$

ce qui amène à la courbe de récession suivante :

$$Q(t) = Q_0 e^{-\alpha_1 t} + \frac{\alpha_2 Q_1}{\alpha_2 - \alpha_1} (e^{-\alpha_1 t} - e^{-\alpha_2 t}) \quad (\text{II.37})$$

Où Q_0 est le débit initial, α_1 et α_2 des constantes, et S_2 le stockage du réservoir. Ce modèle est utilisé dans [Moore, 1997] pour caractériser un système composé de deux zones: une zone avale dans laquelle le réseau hydrographique se développe pendant les orages et où l'écoulement de subsurface revient en surface et ruisselle (exfiltration), et une zone amont qui alimente la première. La première zone correspond à la réponse rapide, la seconde à la réponse lente.

Une autre équation de stockage :

$$Q = \alpha_1 S_1 + \alpha_2 S_2 \quad (\text{II.38})$$

permet de retrouver le modèle à deux réservoirs indépendants parallèles ayant chacun leur propre cinétique de vidange cette fois :

$$Q(t) = Q_1 e^{-\alpha_1 t} + Q_2 e^{-\alpha_2 t} \quad (\text{II.39})$$

Ce modèle est utilisé dans [Clausen, 1992] pour caractériser des aquifères indépendants superposés séparés par un aquitard, et dans [James and Thompson, 1970] pour un débit ayant deux composantes : un débit de base et un écoulement de subsurface. Cette fonction fait également partie des équations utilisées pour décrire le fonctionnement des aquifères karstiques.

Pour finir, il convient de citer également le noyau d'Hayami, développé par [Hayami, 1951] et remis au goût du jour par [Moussa, 1996]. Ce modèle est une solution de l'équation de l'onde diffusante, elle-même issue des équations de Saint Venant. Le noyau de ce modèle peut être utilisé comme fonction de transfert [Charlier et al., 2008] sous la forme :

$$H(t) = \left(\frac{\omega \cdot z}{\pi} \right)^{\frac{1}{2}} \cdot \frac{\exp\left(-\frac{z}{\omega} \frac{t}{t}\right)}{(t)^{3/2}} \quad (\text{II.40})$$

Les deux paramètres de la fonction ω et z sont respectivement le décalage temporel de la fonction et un paramètre de forme. À noter que la forme de cette équation est comparable à la solution d'une équation de convection dispersion 1-D milieu semi-infini, c'est-à-dire une distribution gaussienne de moyenne ω et où z est une composition de la variance.

II.3.2 Courbes de récession des aquifères karstiques

L'étude des aquifères karstiques a également généré bon nombre de fonctions de transfert car l'analyse des courbes de récession des sources et des rivières associées à ces formations tient une place importante en hydrogéologie. [Drogue, 1972] par exemple modélise le débit de sources karstiques à l'aide d'une fonction hyperbolique :

$$Q(t) = \frac{Q_0}{(1 + \alpha t)^n} \quad (\text{II.41})$$

Cette fonction empirique est une version généralisée de la formule de Boussinesq (que l'on retrouve pour $n=2$) permettant également de retrouver d'autres modèles empiriques et conceptuels alors utilisés, selon les différentes valeurs de n .

[Mangin, 1975] utilise un modèle composé de deux fonctions : φ représentant le débit des zones saturées avec un réservoir linéaire, et ψ une fonction homographique empirique représentant le débit des zones non-saturées.

$$Q(t) = \varphi(t) + \psi(t) \quad (\text{II.42})$$

avec

$$\varphi(t) = q_{r0} e^{-\alpha t} \quad (\text{II.43})$$

et

$$\psi(t) = (q_{r0} - Q_0) \frac{1 - \eta t}{1 + \epsilon t} \quad (\text{II.44})$$

où q_{r0} est la valeur extrapolée à $t=0$ du débit de base, η et ϵ sont des constantes caractéristiques du système.

Toujours à propos des sources karstiques, [Padilla et al., 1994] utilisent une version modifiée de l'équation de Coutagne :

$$Q(t) = (Q_0 - Q_c)[1 + (n-1)\alpha t]^{n/(1-n)} + Q_c \quad (\text{II.45})$$

où Q_c est un débit constant, pouvant provenir selon les auteurs d'un aquitard extérieur au système karstique ou des zones peu transmissives de l'aquifère. On retrouve l'équation (II.29) quand Q_c tend vers zéro.

Dans [Griffiths and Clausen, 1997], afin de représenter la vidange de conduits karstiques ou des écoulements de subsurface à faible débit, les auteurs utilisent un modèle basé sur la vidange d'un réservoir à travers un orifice libre (contrairement au réservoir linéaire où l'orifice est poreux) :

$$Q(t) = a - bt \quad (\text{II.46})$$

où a et b sont des constantes.

Dans le logiciel TEMPO du BRGM [Pinault, 2007], cinq FTP sont utilisés afin de caractériser la réponse lente des aquifères karstiques. Ces cinq fonctions sont issues de modèles conceptuels associant réservoirs linéaires et conduits, associé en série ou en parallèle. Les modèles sont les suivants :

- Le modèle de Nash, associant n réservoirs linéaires identiques de coefficient de récession k :

$$H(t) = \frac{1}{k\Gamma(n)} \left(\frac{t}{k}\right)^{n-1} e^{-t/k} \quad (\text{II.47})$$

- Le modèle de Zoch, associant un réservoir linéaire de coefficient de récession k et un conduit en série :

$$\begin{cases} H(t) = \frac{1}{T}(1 - e^{-t/k}) & \text{pour } t < T \\ H(t) = \frac{1}{T}(1 - e^{-T/k})e^{-(t-T)/k} & \text{pour } t \geq T \end{cases} \quad (\text{II.48})$$

- Le modèle à deux réservoirs indépendants en parallèle ayant chacun leur propre cinétique de vidange (coefficient de récession k_1 et k_2), déjà présenté au paragraphe précédent sous une forme légèrement différente (II.39) :

$$H(t) = \frac{\theta}{k_1} e^{-t/k_1} + \frac{1-\theta}{k_2} e^{-t/k_2} \quad (\text{II.49})$$

- Le modèle à un réservoir et deux coefficients de récession k_1 et k_2 dont le fonctionnement change au bout de T jours :

$$\begin{cases} H(t) = \frac{e^{-t/k_2}}{k_2(e^{-t/k_2} - 1) + k_1} e^{-t/k_2} & \text{pour } t < T \\ H(t) = \frac{e^{-t/k_1}}{k_2(e^{-t/k_2} - 1) + k_1} e^{-t/k_1} & \text{pour } t \geq T \end{cases} \quad (\text{II.50})$$

- Et enfin le modèle de Samani et Ebrahimi [Samani and Ebrahimi, 1996] dont la contribution entre les deux réservoirs est θ et $1-\theta$:

$$H(t) = (1-\theta) \left[1 + (n-1) \frac{t}{k_1} \right]^{n/(1-n)} + \theta e^{-t/k_2} \quad (\text{II.51})$$

II.3.3 Fonctions composites

Certaines fonctions de transfert parmi celles utilisées dans la littérature correspondent à des fonctions plus simples mises en série (équation (II.14)) ou en parallèle (équation (II.15)). D'autres fonctions de transfert existent, mettant en série des fonctions qui ne sont pas des courbes de récession. Elles seront regroupées dans ce travail sous le terme de fonctions composites.

[Pinault, 2007] introduit, en plus des cinq FTP utilisées exclusivement pour la réponse « lente » des aquifères karstiques et présentées précédemment, une fonction de transfert plus générale, utilisée pour la réponse « rapide ». Elle correspond au produit de convolution d'une gaussienne par une exponentielle :

$$H(t) = A \cdot e^{-\ln(2) \frac{(t-T)^2}{D^2}} * e^{-\ln(2) \frac{t}{L}} \quad (\text{II.52})$$

où A est le facteur de normalisation, T le retard du processus de recharge par rapport à l'évènement pluvieux (i.e. la position de la gaussienne), D la durée du phénomène (i.e. la largeur à mi-hauteur de la gaussienne), et L le temps de demi tarissement. D'après son auteur, la partie gaussienne de la fonction traduit la recharge du réservoir alors que la partie exponentielle traduit la vidange linéaire du réservoir.

[Singh et al., 2007] utilisent la fonction de transfert de leur modèle dit « hybride étendu » (« *extended hybrid model* ») pour une étude portant également sur des réponses aux temps courts. Ce modèle conceptuel est formé d'une succession de n entités, chacune constituée de deux réservoirs linéaires de coefficient de stockage k_1 et k_2 , séparés par un conduit induisant un retard T . La fonction de transfert de ce modèle est :

$$H(t) = \left(\frac{(t-nT)}{(n-1)!} \right) \left[\frac{e^{-\frac{(t-nT)}{k_1}}}{(k_1-k_2)^n} + \frac{e^{-\frac{(t-nT)}{k_2}}}{(k_2-k_1)^n} \right] \quad (\text{II.53})$$

À partir de cette équation, on peut retrouver la FTP du modèle de Nash en prenant $k_2 = T = 0$, $k_1 = k$ et $n = 2$, ainsi que la FTP du modèle dit « hybride » (« *hybrid model* ») formé d'une succession de n blocs constitués de deux réservoirs linéaires en série. Pour $n = 2$, la forme générale de la fonction de transfert du modèle hybride vaut :

$$H(t) = \frac{1}{(k_1-k_2)^2} \left[\left(t e^{-\frac{t}{k_1}} + t e^{-\frac{t}{k_2}} \right) - \frac{2k_1k_2}{(k_1-k_2)} \left(e^{-\frac{t}{k_1}} - e^{-\frac{t}{k_2}} \right) \right] \quad (\text{II.54})$$

II.3.4 Fonctions de densité de probabilités

Une autre approche fréquemment utilisée dans la génération d'hydrographe unitaire est l'utilisation de fonctions de densité de probabilités (FDP). En théorie des probabilités, ces

fonctions représentent la probabilité relative d'une variable aléatoire de prendre une certaine valeur. Pour une fonction continue f définie sur $[a, b]$, la FDP f' est égale à :

$$f'(x) = \frac{f(x)}{\int_a^b f(x)dx} \quad (\text{II.55})$$

Cette définition reste générale et les FDP sont souvent solutions d'équations formalisant de multiples processus physiques, et dont l'intégrale est égale à 1. Bien qu'à l'origine les FDP aient été utilisées en hydrologie de manière empirique du fait de leur allure similaire à celles des hydrographes unitaires, ces fonctions peuvent avoir un sens physique réel. Par exemple, une gaussienne traduit la probabilité de présence d'une particule (une fraction de masse ou de volume) en un point dans un processus de diffusion. Une exponentielle décroissante traduira la probabilité des temps de transition par diffusion au travers d'une interface, etc. Lors d'une étude sur la réponse aux temps courts de bassins versants non-jaugés, [Bhunya et al., 2007] utilisent les quatre FDP suivantes :

La FDP de la distribution Gamma Γ , équivalente au modèle de Nash, équation (II.47) :

$$H(t) = \frac{1}{k\Gamma(n)} \left(\frac{t}{k}\right)^{n-1} e^{-t/k} \quad (\text{II.56})$$

La FDP de la distribution Khi carré χ^2 :

$$H(t) = \frac{1}{2^{k/2}\Gamma(k/2)} t^{(k/2)-1} e^{-t/2} \quad (\text{II.57})$$

La FDP de la distribution Bêta β (que les auteurs recommandent d'utiliser, notamment en raison de sa flexibilité) :

$$H(t) = \frac{1}{\beta(a,b)} t^{a-1} (1-t)^{b-1} \quad (\text{II.58})$$

où $\beta(a,b)$ est la fonction Beta définie par $\beta(a,b) = \int_0^1 x^{a-1} (1-x)^{b-1} dx$.

La FDP de la distribution de Weibull :

$$H(t) = \frac{a}{b} \left(\frac{t}{b}\right)^{b-1} e^{-(t/b)^a} \quad (\text{II.59})$$

où a , b , n et k sont les différents paramètres des FDP à optimiser. Ce sont des FDP de la sorte qui sont utilisées dans la bibliothèque de fonction de MASH.

II.3.5 Bibliothèque de fonction de MASH

On rappelle que MASH permet de choisir chacune des fonctions de transfert dans une bibliothèque de fonctions. Cette bibliothèque est pour l'instant assez réduite car elle ne comporte que quatre fonctions de transfert mais étant donnée l'architecture du modèle, ajouter de nouvelles fonctions ne pose aucun problème majeur du point de vue du développement logiciel. Après avoir réalisé qu'il n'existait pas dans la littérature de relation bijective entre la FTP utilisée et l'hydrosystème représenté (i.e. une même fonction de transfert peut modéliser plusieurs phénomènes, et la réponse d'hydrosystèmes de même nature peut être approchée par plusieurs fonctions), le choix a été fait de sélectionner des fonctions de transfert utilisables selon trois critères : leur faible paramétrisation, leur caractère général et leur interprétation physique.

La faible paramétrisation permet au modèle global d'être parcimonieux et d'éviter « la malédiction de la dimension » lors de la procédure d'optimisation. Les fonctions restent simples et la complexité provient uniquement de la mise en série/parallèle de fonctions élémentaires. Le sens physique de cette mise en série (événements successifs dans le temps ou l'espace) ou parallèle (événements concomitants ou disjoints spatialement) est simple et relativement facile à jauger au regard de la réalité du problème modélisé. Le caractère général représente la capacité d'une fonction de transfert à englober d'autres modèles plus simples selon la valeur de ses paramètres. Une même fonction représente ainsi plusieurs modèles, ce qui facilite la programmation du logiciel et augmente les possibilités offertes. L'interprétation physique apporte les avantages des modèles conceptuels à la modélisation systémique comme l'interprétation phénoménologique du fonctionnement du système et l'éventuel rapprochement entre les paramètres du modèle et les caractéristiques physiques du terrain.

Les quatre fonctions de la bibliothèque de MASH sont toutes des FDP : loi Gamma, loi normale, loi Bêta et loi de puissance (les fonctions ci-dessous ne sont pas toutes normalisées par un coefficient multiplicateur qui amènerait leur intégrale à un. On verra plus loin que ce coefficient peut être assimilé dans le gain de la branche du modèle) :

- Loi Gamma

$$H(t) = \frac{1}{k\Gamma(n)} \left(\frac{t}{k}\right)^{n-1} e^{-t/k} \quad (\text{II.60})$$

avec

$$\Gamma(n) = \int_0^{\infty} t^{n-1} e^{-t} dt \quad (\text{II.61})$$

- Loi normale

$$H(t) = e^{-\frac{(t-T)^2}{\sigma^2}} \quad (\text{II.62})$$

- Loi Bêta

$$H(t) = \frac{t^{n-1}(1-t)^{m-1}}{\beta(n, m)} \quad (\text{II.63})$$

avec
$$\beta(x, y) = \int_0^1 t^{x-1}(1-t)^{y-1} dt \quad (\text{II.64})$$

- Loi de puissance

$$H(t) = \frac{1}{(1 + \alpha t)^n} \quad (\text{II.65})$$

Certaines de ces fonctions possèdent normalement un paramètre de plus qui est un coefficient multiplicateur. Celui-ci n'est pas pris en compte pour les fonctions élémentaires de transfert et est remplacé par le gain total de la branche afin de limiter le nombre de paramètres à optimiser. D'un point de vue modélisation, il est équivalent d'avoir un gain par fonction de transfert élémentaire et un gain pour la fonction de transfert équivalente (le produit de convolution des trois fonctions élémentaires en série). En effet une fonction de transfert peut s'écrire sous la forme :

$$H^\alpha(t) = K^\alpha \cdot H'^\alpha(t) \quad (\text{II.66})$$

où K^α est le gain de la fonction de transfert. S'il existe une fonction de transfert équivalente $H^{eq}(t)$ égale au produit de convolution de n fonctions en série des trois fonctions de transfert en série :

$$H^{eq}(t) = H^1 * H^2 * \dots * H^n(t) \quad (\text{II.67})$$

alors
$$H^{eq}(t) = (K^1 \cdot H^1) * (K^2 \cdot H^2) * \dots * (K^n \cdot H^n)(t) \quad (\text{II.68})$$

Trivialement, les gains K ne dépendant pas des variables d'intégrations des produits de convolutions, le gain équivalent $K^{eq} = \prod_{i=1}^n K^i$ vérifie :

$$H^{eq}(t) = K^{eq} \cdot (H^1 * H^2 * \dots * H^n(t)) \quad (\text{II.69})$$

Les fonctions de la bibliothèque n'ont donc que deux paramètres.

Une cinquième fonction de transfert est disponible à titre expérimental dans MASH, uniquement en mode manuel. Cette fonction n'est pas une FDP comme les quatre autres fonctions présentées précédemment mais la solution analytique approchée d'un problème d'advection et de changement de phase, également connu sous le nom d'équation du télégraphe (F. Delay, communication personnelle, 4 mai 2012).

$$H(t) = \frac{x}{\frac{u}{R}t(2\pi\langle\tilde{r}(t).\tilde{r}(t)\rangle)^{\frac{1}{2}}} \exp\left(-\frac{\left(x-\frac{u}{R}t\right)^2}{2\langle\tilde{r}(t).\tilde{r}(t)\rangle}\right) \quad (\text{II.70})$$

avec
$$R = \frac{\lambda + \mu}{\mu} \quad (\text{II.71})$$

et
$$\langle\tilde{r}(t).\tilde{r}(t)\rangle = \frac{2\lambda\mu u^2 t}{(\lambda + \mu)^3} (1 - \exp(-(\lambda + \mu)t)) - \frac{2\lambda\mu u^2}{(\lambda + \mu)^4} \Gamma((\lambda + \mu)t) \quad (\text{II.72})$$

Même si cette fonction de transfert n'est pas utilisée dans la suite des travaux présentés ici, elle permet d'obtenir des résultats préliminaires prometteurs, et mériterait d'être étudiée plus en détail dans une étude ultérieure.

La première utilisation de la distribution Gamma [Nash, 1957] était motivée par sa ressemblance avec un hydrographe unitaire. D'un point de vue théorique, cette fonction simule le comportement de n réservoirs linéaires identiques en série (Figure II-8), le paramètre n pouvant au demeurant ne pas être entier. L'équation (II.60) se retrouve aisément. La fonction de transfert d'un réservoir linéaire est égale à

$$h(t) = \frac{1}{k} e^{-t/k} \quad (\text{II.73})$$

soit dans le domaine de Laplace

$$H(p) = \mathcal{L}\{h(t)\} = \frac{1}{1+kp} \quad (\text{II.74})$$

Comme le produit de convolution dans le domaine temporel équivaut à la multiplication scalaire dans le domaine de Laplace, le fait d'avoir n fonctions de transfert en série (donc convoluées) s'écrit simplement

$$H_n(p) = \left(\frac{1}{1+kp}\right)^n \quad (\text{II.75})$$

Et donc en revenant dans le domaine temporel, on retrouve l'équation (II.60) :

$$h_n(t) = \mathcal{L}^{-1}\{H_n(p)\} = \frac{1}{k\Gamma(n)} \left(\frac{t}{k}\right)^{n-1} e^{-t/k} \quad (\text{II.76})$$

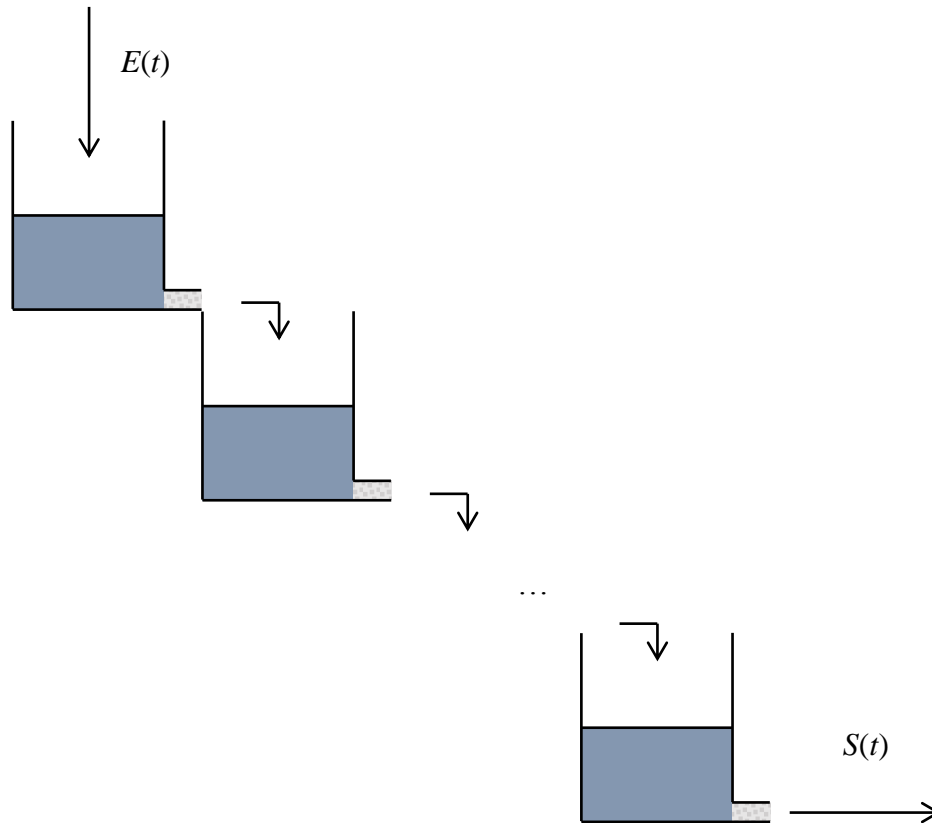


Figure II-8 : Schéma de fonctionnement du modèle de Nash

La FDP de la loi Gamma est largement utilisée en hydrologie pour simuler des débits [Besbes and de Marsily, 1984; Haktanir and Sezen, 1990; Bhunya et al., 2007; Singh et al., 2007] ou encore des hauteurs piézométriques, la solution analytique des écoulements darcéens convergents étant une loi Gamma [von Asmuth et al., 2002]. [Cudennec et al., 2004] ont démontré grâce à la géomorphologie qu'il existait une structure en loi Gamma sous-jacente à l'organisation des réseaux hydrauliques (longueurs hydrauliques et nombres de Strahler), apportant une preuve que l'hydrogramme unitaire peut sous certaines simplifications être approché par cette FDP. Cette constatation enrichit le sens physique donné à la loi Gamma en sus de l'explication plus conceptuelle des n réservoirs linéaires en série.

La deuxième fonction de transfert disponible dans MASH est la loi normale. Cette loi est l'une des plus utilisées afin de représenter les phénomènes naturels. Dans le cadre de la modélisation hydrologique, elle représente l'élargissement de la réponse impulsionnelle dû aux effets dispersif et diffusif. Le temps T mesure alors le temps moyen de résidence de la pluie avant d'arriver à l'exutoire, tandis que σ caractérisera d'autres propriétés du bassin versant. De plus, quand le paramètre σ tend vers 0, la fonction de transfert est équivalente à un Dirac décalé de T ce qui permet : 1- de ne jamais être néfaste au modèle, car si la fonction élémentaire de transfert est inutile, elle sera égale à un Dirac après l'inversion 2- de simuler un retard pur, c'est-à-dire une fonction de transfert correspondant à mouvement convectif pur, encore appelé couramment "effet piston".

La troisième fonction de MASH est la loi Bêta. Elle est utilisée dans un large éventail de problèmes pour décrire le comportement de certaines variables aléatoires et, en hydrologie, pour sa capacité à reproduire des hydrogrammes unitaires [Haktanir and Sezen, 1990; Bhunya et al., 2007]. Si la version standard de la fonction utilisée dans MASH comporte deux paramètres, la version de la fonction utilisée dans [Bhunya et al., 2007] en comporte trois :

$$H(t) = \frac{t^{r-1}(b-t)^{p-r-1}}{\beta(r, p-r)b^{p-1}} \quad (\text{II.77})$$

C'est cette fonction qui donne d'ailleurs les meilleurs résultats grâce à sa « flexibilité » dans cette étude qui compare quatre distributions afin d'identifier celles qui reproduisent le mieux l'hydrogramme de plusieurs bassins versants non jaugés. L'étude comparative de [Haktanir and Sezen, 1990] montre cependant que les deux versions de la fonction (à deux ou trois paramètres) donnent des résultats comparables et qu'il est donc préférable d'utiliser la fonction la plus simple dans le cas général. C'est donc le choix qui a été fait pour l'implémentation dans MASH. La distribution Bêta, comme la distribution Gamma, est utilisée principalement en raison de sa forme se rapprochant de celle des hydrogrammes unitaires. En effet, la forme de ces distributions varie de symétrique à asymétrique positive (i.e. la moyenne est supérieure au mode).

La quatrième fonction de la bibliothèque est la distribution loi de puissance. Elle est incorporée à la bibliothèque pour deux raisons. Premièrement, elle permet de retrouver un grand nombre d'autres fonctions de transfert, notamment parmi celles présentées précédemment (**Tableau II-1**). Elle englobe également un grand nombre de modèles avec seulement deux paramètres. Deuxièmement, alors que les autres lois ont une décroissance plutôt brutale, la loi de puissance permet d'obtenir en jouant sur l'exposant de puissance tout un panel de décroissance, de raide à plate, de la densité de probabilité, ce qui peut s'avérer avantageux par exemple pour la modélisation d'aquifères dont les temps de réponse sont particulièrement longs.

Grâce à ces quatre fonctions et à une architecture appropriée, il est possible de retrouver dix-sept des fonctions vues précédemment (**Tableau II-1**).

Tableau II-1: Associations de fonctions de transfert de la bibliothèque de MASH et modèles qu'elles permettent de retrouver

Fonctions de transfert	Équations des modèles correspondants
loi Gamma	(II.16) ; (II.47) ; (II.56)
loi Bêta	(II.58)
loi de puissance	(II.17) ; (II.19) ; (II.24) ; (II.26) ; (II.29) ; (II.32) ; (II.41)
Deux lois Gamma en parallèle	(II.39) ; (II.49)
loi Gamma en série avec loi normale	(II.48)* ; (II.52)
loi Gamma en parallèle avec loi de puissance	(II.51)
loi Gamma en série avec loi Gamma et loi normale	(II.54)*

* : pour ces modèles, la loi normale est utilisée dans un cas limite et correspond à un retard pur

II.4 Inversion

Les branches et les fonctions de transfert possédant respectivement un et deux paramètres, les modèles créés avec MASH comportent de trois (pour un modèle à une branche et une FT) à vingt-huit paramètres (pour un modèle à quatre branches et quatre FT) à optimiser. Lors des premières étapes du développement de MASH, ces paramètres étaient optimisés manuellement par approche essais-erreurs. Cette méthode d'optimisation bien qu'intéressante est néanmoins limitée notamment à cause de l'équifinalité possible entre les différents paramètres. Bien qu'elle soit toujours disponible afin d'effectuer une approximation grossière des paramètres en première approche, on lui préfère généralement des algorithmes d'optimisation automatique, plus rapides et plus à même de déterminer une solution optimale [Poeter and Hill, 1997]. Les différents paramètres de MASH sont donc inversés grâce à deux algorithmes : PEST (pour « Parameter ESTimation »), une méthode de type Levenberg-Marquardt, et PSO (pour « Particle Swarm Optimization »), une métaheuristique basée sur l'intelligence en essaim. Le principal avantage de tels algorithmes d'inversion est qu'ils ne sont pas intrusifs et ne nécessitent donc pas de modification du code interne du modèle pour fonctionner efficacement (quoique, les sensibilités nécessaires à l'algorithme de Levenberg-Marquardt, gagneraient à être dérivées analytiquement au sein du problème direct). Comme ces méthodes d'inversion sont indépendantes du problème, elles sont utilisées dans de nombreuses disciplines ce qui permet d'avoir une description extensive des algorithmes dans la littérature (comme c'est le cas pour la PSO) voire même des logiciels disponibles gratuitement et adaptables au modèle (comme c'est le cas pour PEST). De plus, lors de développements ultérieurs de MASH, comme l'ajout de fonctions de transferts ou un changement de la fonction de production, aucune modification des solveurs ne serait nécessaire.

Le modèle MASH utilise des données d'entrée pour simuler des données de sortie (**Figure II-9**). La fonction objectif fournit une mesure de l'ajustement entre les sorties modélisées et mesurées. La fonction objectif intégrant la sortie modélisée, elle dépend par conséquent des paramètres des fonctions de transfert à optimiser. Le solveur a lui pour fonction de rechercher le jeu de paramètres auquel correspond la valeur optimale de la fonction objectif (**Figure II-10**). Pour cela, le solveur envoie en entrée du modèle un ou plusieurs jeux de paramètres, récupère la valeur de la fonction objectif correspondante, puis met à jour les jeux de paramètres selon une logique propre à l'algorithme considéré. Le processus est répété itérativement jusqu'à ce qu'un critère d'arrêt soit atteint. Ce critère d'arrêt correspond généralement à un nombre maximum d'itérations ou une amélioration de la valeur de la fonction objectif entre deux itérations jugées trop faible.

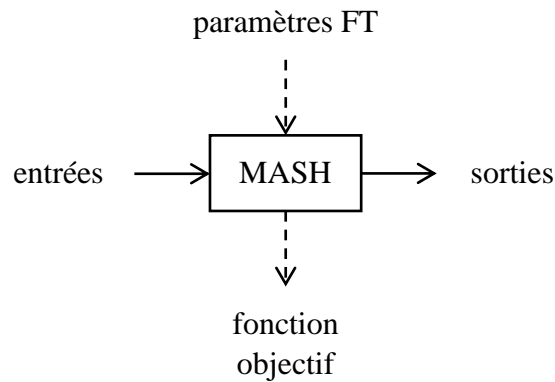


Figure II-9 : Le modèle MASH reçoit des données d'entrée et produit des données de sorties. Les paramètres du modèle à optimiser sont les paramètres des fonctions de transfert (FT). La qualité de l'ajustement pour un jeu de paramètre donné est mesurée par une fonction objectif.

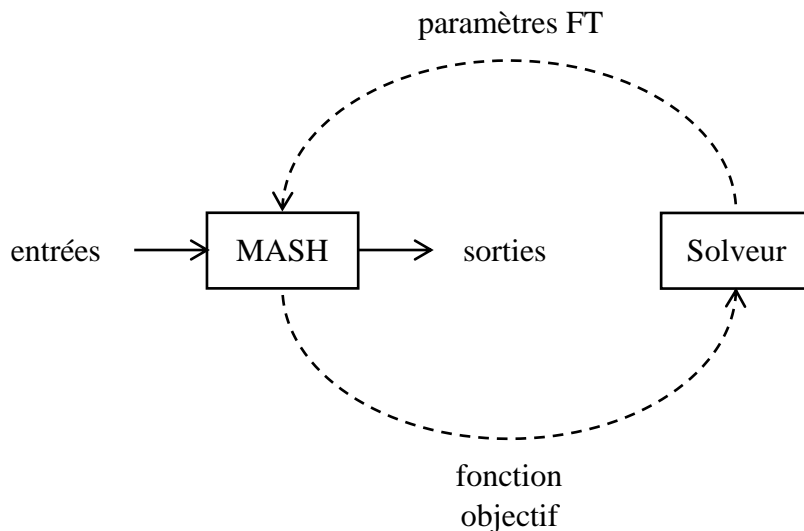


Figure II-10 : Le solveur envoie des jeux de paramètres à MASH qui lui renvoie la valeur de la fonction objectif. Le solveur calcule alors de nouveaux jeux de paramètres jusqu'à ce qu'un critère d'arrêt soit atteint.

II.4.1 PEST (Parameter ESTimation)

Le premier algorithme d'optimisation automatique à avoir été utilisé lors du développement MASH est PEST, un solveur non-intrusif développé par John Doherty, largement utilisé en modélisation hydrogéologique [Ginn *et al.*, 2007] qui prend la forme d'une série de fichiers exécutables Windows. Le développement de PEST s'étant étalé sur presque deux décennies, toutes les subtilités de calcul ne seront pas évoquées dans la présentation qui suit. Pour une

description plus détaillée, il est possible de se référer au manuel utilisateur de PEST [Doherty, 2010] dont les explications qui suivent sont en partie tirées.

Comme on l'a vu au Chapitre I : , un système linéaire peut être décrit simplement par l'équation linéaire :

$$\mathbf{y} = \mathbf{A}\mathbf{p} \quad (\text{II.78})$$

avec \mathbf{A} la matrice des relations linéaires entre modèle et paramètres, \mathbf{p} le vecteur des paramètres et \mathbf{y} le vecteur des variables de sortie du modèle. Dans le cas général, les données simulées par le modèle sont différentes des observations qu'on essaie de reproduire. Une manière de définir le jeu de paramètre optimal est de choisir celui qui minimise la somme du carré des résidus entre la sortie générée par le modèle et les observations. Cela revient à minimiser une fonction objectif Φ définie par :

$$\Phi(\mathbf{p}) = (\mathbf{A}\mathbf{p} - \hat{\mathbf{y}})^T (\mathbf{A}\mathbf{p} - \hat{\mathbf{y}}) \quad (\text{II.79})$$

avec $\hat{\mathbf{y}}$, le vecteur des observations. On peut démontrer que le vecteur \mathbf{p} qui minimise Φ satisfait l'équation normale :

$$\mathbf{p} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \hat{\mathbf{y}} \quad (\text{II.80})$$

Le solveur de PEST ajoute une matrice carré diagonale \mathbf{Q} afin d'ajouter un poids aux différentes observations, dans le cas où celles-ci représenteraient par exemple des quantités d'ordre de grandeur différent. L'équation (II.79) devient alors :

$$\Phi(\mathbf{p}) = (\mathbf{A}\mathbf{p} - \hat{\mathbf{y}})^T \mathbf{Q} (\mathbf{A}\mathbf{p} - \hat{\mathbf{y}}) \quad (\text{II.81})$$

Le vecteur qui minimise Φ , i.e. le meilleur estimateur linéaire devient alors :

$$\mathbf{p} = (\mathbf{A}^T \mathbf{Q} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{Q} \hat{\mathbf{y}} \quad (\text{II.82})$$

Cependant, la plupart des modèles ne sont pas linéaires et il est donc impossible de lier paramètres et observations à l'aide d'une relation comme celle décrite par l'équation (II.78). La linéarisation des modèles s'avère alors nécessaire afin d'utiliser la théorie présentée ci-dessus pour estimer leurs paramètres.

Soit \mathbf{M} l'opérateur direct transformant un jeu de paramètre \mathbf{p}_0 en variables de sortie \mathbf{y}_0 :

$$\mathbf{y}_0 = \mathbf{M}(\mathbf{p}_0) \quad (\text{II.83})$$

Si \mathbf{M} est continument différentiable (i.e. de classe \mathcal{C}^1), pour un vecteur de paramètres \mathbf{p} suffisamment proche de \mathbf{p}_0 , il est possible de générer un vecteur de variable de sortie \mathbf{y} grâce au développement de Taylor qui donne au premier ordre l'approximation suivante :

$$\mathbf{y} = \mathbf{y}_0 + \mathbf{J} \cdot (\mathbf{p} - \mathbf{p}_0) \quad (\text{II.84})$$

où \mathbf{J} est la matrice jacobienne de \mathbf{M} , i.e. la matrice des dérivées partielles du premier ordre des variables de sortie par rapport aux paramètres. La fonction objectif de type moindres carrés est alors définie par :

$$\Phi(\mathbf{p}) = (\mathbf{J} \cdot (\mathbf{p} - \mathbf{p}_0) - (\hat{\mathbf{y}} - \mathbf{y}_0))^T \mathbf{Q} \cdot (\mathbf{J} \cdot (\mathbf{p} - \mathbf{p}_0) - (\hat{\mathbf{y}} - \mathbf{y}_0)) \quad (\text{II.85})$$

où $\hat{\mathbf{y}}$ représente le vecteur des observations. L'équation (II.85) est équivalente à l'équation (II.81) et il est donc possible de calculer le vecteur des paramètres actualisé $(\mathbf{p} - \mathbf{p}_0)$ à partir du vecteur $(\hat{\mathbf{y}} - \mathbf{y}_0)$ qui mesure l'écart entre les valeurs calculées par le modèle \mathbf{y}_0 et les observations $\hat{\mathbf{y}}$ de la même manière qu'avec l'équation (II.82). En notant \mathbf{u} le vecteur des paramètres actualisé :

$$\mathbf{u} = (\mathbf{J}^T \mathbf{Q} \mathbf{J})^{-1} \mathbf{J}^T \mathbf{Q} \cdot (\hat{\mathbf{y}} - \mathbf{y}_0) \quad (\text{II.86})$$

L'équation (II.84) est une approximation et donc l'équation (II.86) en est une également. Il n'est donc pas garanti que le nouveau jeu de paramètres \mathbf{p} , égal au jeu de paramètres courant \mathbf{p}_0 auquel on ajoute le vecteur \mathbf{u} , soit celui qui minimise la fonction objectif. Ce jeu de paramètres \mathbf{p} est donc utilisé comme point de départ pour déterminer le vecteur de paramètre actualisé, et ainsi de suite, jusqu'à ce qu'on arrive (peut-être), au minimum global de Φ . Ce processus de convergence itérative est schématiquement représenté en Figure III-11 pour un problème à deux paramètres.

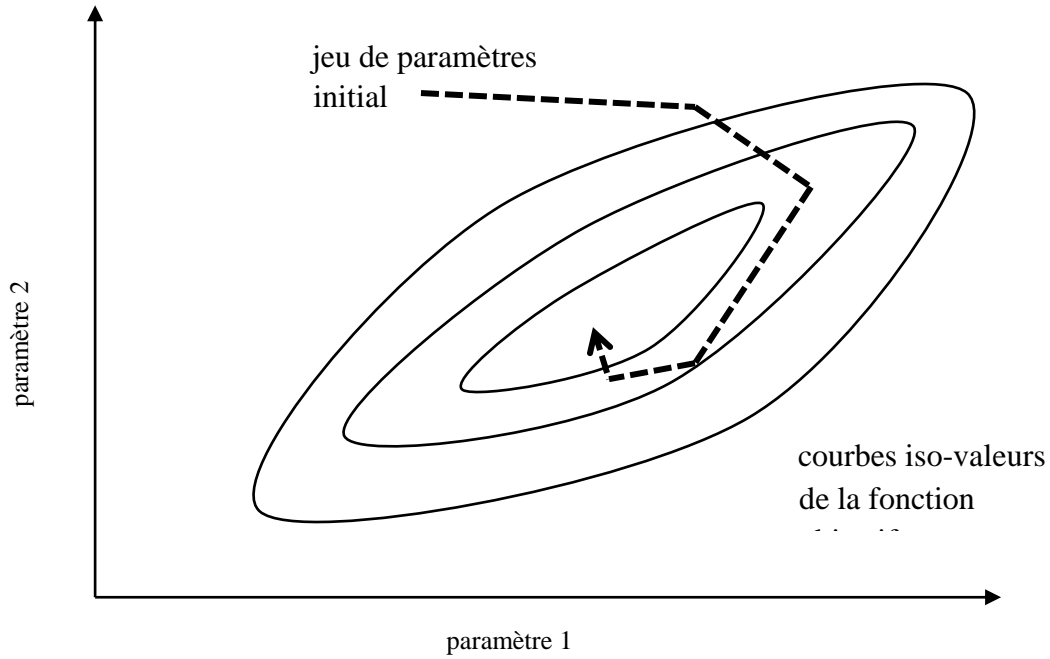


Figure II-11: Amélioration itérative du jeu de paramètre initial vers le minimum global de la fonction objectif (d'après [Doherty, 2010])

L'équation (II.86) peut également s'écrire de la manière suivante :

$$\mathbf{u} = (\mathbf{J}^T \mathbf{Q} \mathbf{J})^{-1} \mathbf{J}^T \mathbf{Q} \mathbf{r} \quad (\text{II.87})$$

où \mathbf{u} est le vecteur des paramètres actualisé et \mathbf{r} est le vecteur des résidus entre simulation et observation à l'itération précédente.

Pour des méthodes de type gradient (« *steepest descent* »), la mise à jour des paramètres n'est pas faite selon la direction du vecteur \mathbf{u} mais selon la direction du vecteur \mathbf{g} , égal au gradient de la fonction objectif Φ dans l'espace des paramètres, dans le sens négatif. La $i^{\text{ème}}$ coordonnée de \mathbf{g} est égale à :

$$\mathbf{g}_i = \frac{\partial \Phi}{\partial \mathbf{p}_i} \quad (\text{II.88})$$

c'est-à-dire la dérivée partielle de la fonction objectif par rapport au $i^{\text{ème}}$ paramètre. Le vecteur gradient \mathbf{g} peut s'écrire sous la forme :

$$\mathbf{g} = -2\mathbf{J}^T \mathbf{Q} \mathbf{r} \quad (\text{II.89})$$

À noter que bien que la direction du vecteur $-\mathbf{g}$ soit celle de la pente la plus forte, il est souvent préférable d'utiliser le vecteur \mathbf{u} , en particulier dans le cas où les paramètres sont corrélés. En effet, pour ce genre de problème, le fait de suivre la pente la plus forte peut conduire au

phénomène de « zigzag » (« *hemstitching* »), le jeu de paramètres passant de part et d'autre de la vallée de la fonction objectif lors des itérations successives (Figure II-12).

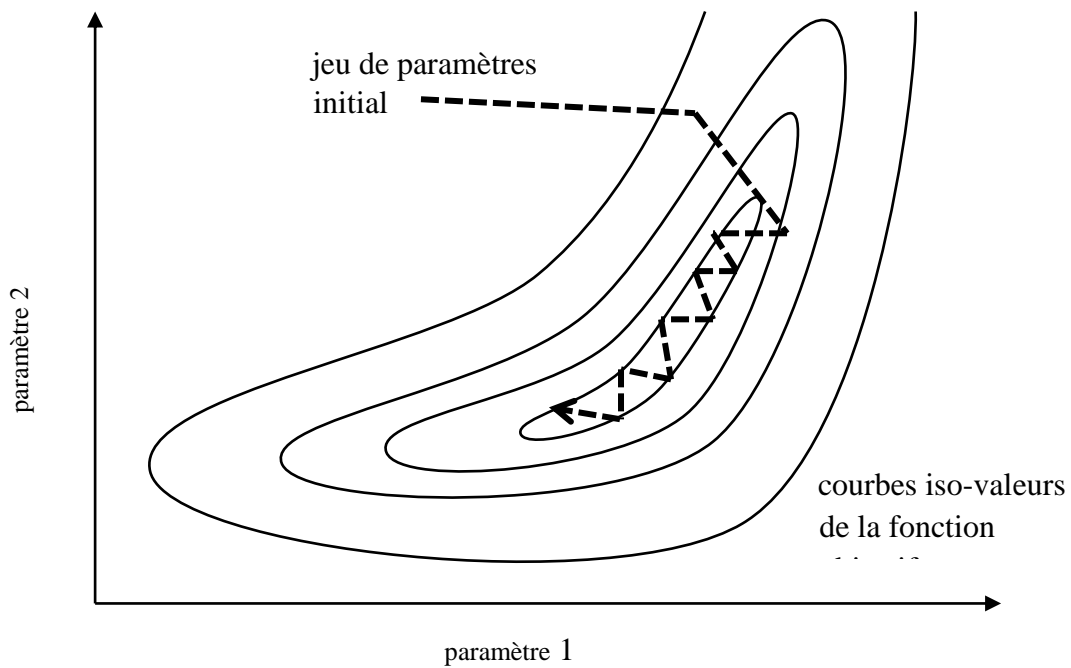


Figure II-12 : Phénomène de zigzag (« *hemstitching* ») lors de l'optimisation (d'après [Doherty, 2010])

Toutefois, lors des premières itérations, il est intéressant de décaler \mathbf{u} vers $-\mathbf{g}$ afin d'accélérer la convergence lors des premiers calculs. Pour cela, un paramètre α qui décroît lors des itérations successives est ajouté à l'équation (II.87) qui devient :

$$\mathbf{u} = (\mathbf{J}^T \mathbf{Q} \mathbf{J} + \alpha \mathbf{I})^{-1} \mathbf{J}^T \mathbf{Q} \mathbf{r} \quad (\text{II.90})$$

où α est le paramètre de Levenberg-Marquardt du nom des mathématiciens l'ayant découvert puis publié, et \mathbf{I} est la matrice identité.

On constate donc que si α est très grand, la direction de \mathbf{u} s'approche de celle de $-\mathbf{g}$, ce qui équivaut à la méthode du gradient. Et si α est nul, on retrouve l'équation (II.87), ce qui équivaut à la méthode de Gauss-Newton. La valeur de α dans PEST est donc relativement grande lors des premières itérations puis décroît au fur et à mesure que l'on se rapproche de la valeur optimale de Φ .

PEST est lancé directement depuis MASH. Un premier programme charge les données de sortie observées qui permettront de calculer la fonction objectif. Ce même programme crée également un jeu d'instructions permettant le bon fonctionnement de PEST, en l'informant notamment du nombre de paramètres à optimiser. Un deuxième programme démarre ensuite l'optimisateur à proprement dit. PEST crée des jeux de paramètres, exécute MASH grâce à une commande DOS

et récupère les données simulées de sortie. À chaque itération, PEST calcule la matrice jacobienne en exécutant plusieurs simulations (calcul par perturbation sur les paramètres des sensibilités présentes dans la matrice Jacobienne) puis calcule la valeur de la fonction objectif pour différentes valeurs du paramètre α . Le jeu de paramètres qui sert de point de départ à la nouvelle itération est celui qui a donné la valeur minimale de la fonction objectif à l'itération précédente. PEST s'arrête lorsqu'un des nombreux critères d'arrêt est atteint (par exemple lorsque les différences successives de valeurs de la fonction objectif entre itérations sont inférieures à un seuil) et renvoie le jeu de paramètre optimum ainsi qu'une analyse de sensibilité des paramètres et des observations.

PEST est basé sur l'algorithme de Levenberg-Marquardt, qui a déjà fait ses preuves pour l'optimisation des paramètres de modèles systémiques [Cigizoglu and Kisi, 2005]. L'expérience suivante a été menée afin de vérifier que PEST convenait à l'optimisation des paramètres de MASH : un jeu de paramètres prédéfini est utilisé pour créer une chronique de sortie simulée et cette chronique est ensuite utilisée comme chronique d'observation. Il est donc théoriquement possible pour PEST de trouver un jeu de paramètres qui rend nulle la fonction objectif. Cette simulation comporte 3 fonctions de transfert réparties dans 3 branches parallèles (soit 9 paramètres à optimiser) pour 1200 données d'observation. Lors de l'optimisation, l'évolution des paramètres est suivie à chaque itération (**Figure II-13**). Pour les neuf paramètres du modèle, la convergence des paramètres est excellente et au bout des 31 itérations, l'écart relatif entre les paramètres prédéfinis et les paramètres calculés est inférieur à 3% (**Tableau II-2**).

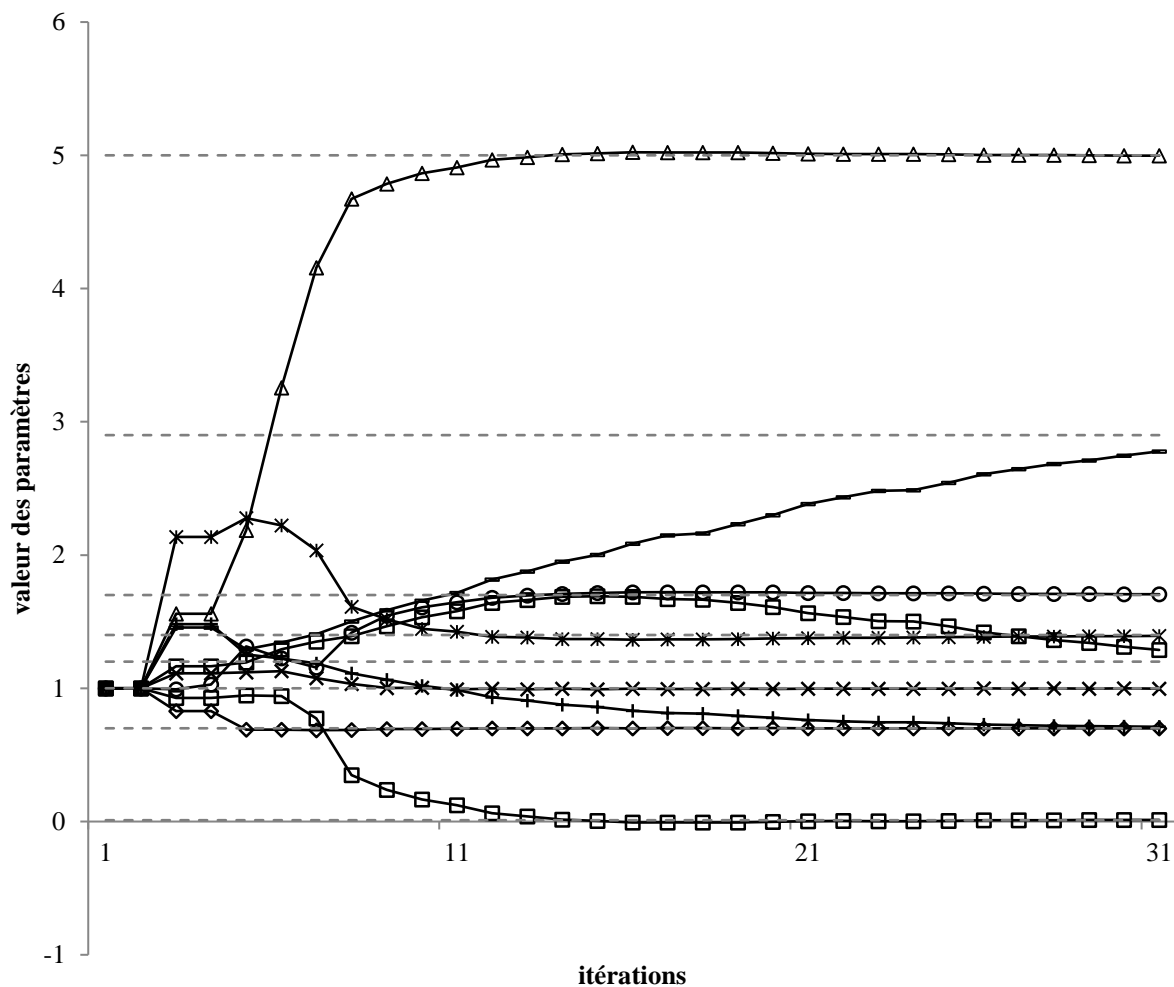


Figure II-13 : Évolution de la valeur des différents paramètres lors de l'optimisation par PEST (traits pleins) et valeur réelle des paramètres (traits pointillés)

Tableau II-2 : Valeurs réelles des paramètres et estimations effectuées par PEST

	Paramètre	Valeur réelle	Estimation PEST
Branche 1	gain	0.7	0.70014
	paramètre 1 loi normale	0.01	0.01292482
	paramètre 2 loi normale	5	4.99814
Branche 2	gain	1	0.998273
	paramètre 1 loi Gamma	1.4	1.39331
	paramètre 2 loi Gamma	1.7	1.70549
Branche 3	gain	0.7	0.710908
	paramètre 1 loi normale	1.2	1.28728

La fonction objectif à la fin de l'optimisation est égale à 0.007967. À titre de comparaison, la valeur initiale 15402.60, pour tous les paramètres égaux à 1. Le critère d'arrêt est l'atteinte du nombre maximum d'itérations et la durée de l'optimisation est à titre d'information d'environ 858 secondes (pour un processeur Intel Core 2 Duo P8600 cadencé à 2.40GHz). PEST est donc capable d'optimiser convenablement les paramètres de MASH, au moins dans certains cas. On notera que les paramètres des fonctions de transfert ne sont pas des grandeurs variant sur de nombreux ordres de grandeur. Ils se révèlent par conséquent assez sensibles ce qui facilite leur recherche par des algorithmes de descente. En d'autres termes, un logiciel comme PEST doit pouvoir optimiser de nombreux problèmes inverses sur des approches systémiques. Une étude exhaustive dépassant le cadre de ce travail pourrait être envisagée en jouant sur les diverses architectures de MASH (multiplicité des branches et multiplicité des séries de fonctions par branche).

PEST présente cependant une limite liée à la nécessité d'avoir des fonctions continument dérivables. En effet, tous les paramètres de MASH sont définis sur $]0, +\infty[$ à l'exception du

paramètre n de la loi Gamma $\frac{1}{k\Gamma(n)}\left(\frac{t}{k}\right)^{n-1}e^{-t/k}$ qui doit appartenir à $[1, +\infty[$. Pour des raisons

liées à l'interfaçage de PEST avec MASH (au niveau de l'écriture du fichier de configuration PEST par MASH), il est très compliqué d'imposer des espaces de définition différents selon la fonction de transfert utilisée, d'autant que chaque fonction peut utiliser n'importe quelle disposition dans les douze disponibles (4 branches à trois fonctions). Le choix a donc été fait de prendre le domaine de définition le plus large possible, pour tous les paramètres de toutes les fonctions, afin de maximiser les chances de trouver une solution optimale. L'espace des solutions est donc $]0, +\infty[^p$ avec p le nombre de paramètres. Quand le paramètre n de la loi Gamma est compris entre 0 et 1, la valeur de la fonction de transfert à $t=0$ tend vers $+\infty$. Le produit de convolution de l'entrée par la fonction de transfert possède alors à son tour des valeurs égales à $+\infty$ et la fonction objectif devient impossible à calculer pour PEST. Pour pallier cette erreur, l'algorithme modifie la valeur du produit de convolution en remplaçant les valeurs égales à $+\infty$ par 0. Ainsi la fonction objectif est prolongée pour les valeurs du paramètre $n \in]0, 1[$ par une valeur normalement assez grande (**Figure II-14**). PEST arrive néanmoins à identifier une solution optimale quand le modèle ne possède qu'une seule FT loi Gamma. Mais quand le modèle en intègre plusieurs, tous les paramètres restent généralement bloqués à 1, en raison de la discontinuité évoquée ci-dessus et illustrée en **Figure III-14**.

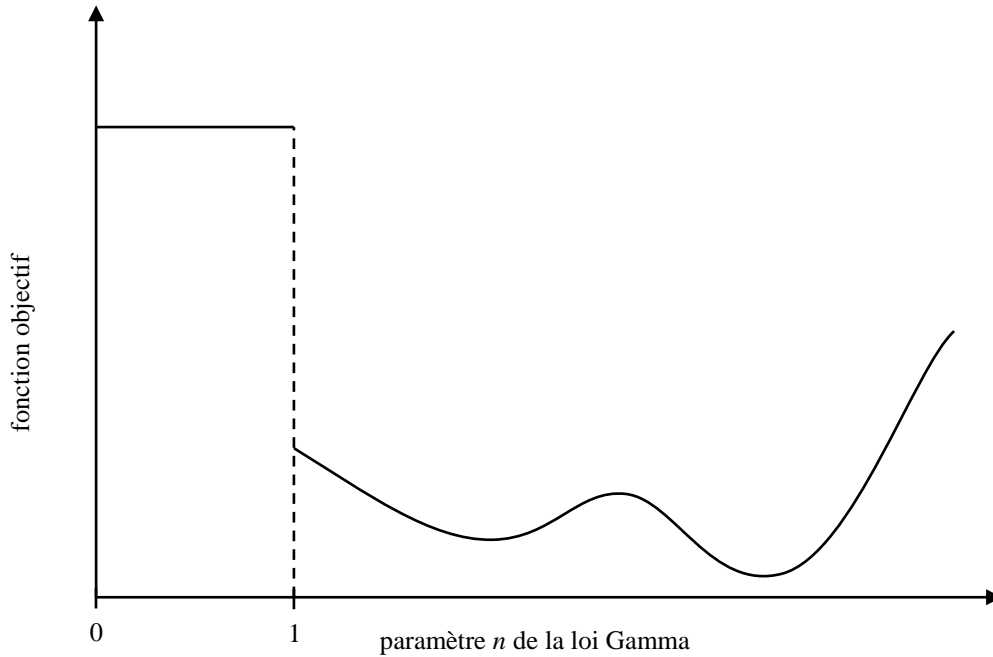


Figure II-14 : Non-continuité de la fonction objectif

Hormis ce problème lié au fait d’avoir une fonction de transfert qui n’est pas de classe \mathcal{C}^1 dans l’espace des paramètres, il existe une autre limite de PEST qui peut se manifester dans l’optimisation. Du fait de la construction même du couplage MASH-PEST, si l’on tente d’optimiser des fonctions de transfert identiques et ayant la même entrée, les paramètres convergent obligatoirement vers des valeurs identiques. Pour illustrer cela, une deuxième expérience a été menée, similaire à la première sauf que cette fois le modèle possède seulement deux branches utilisant deux entrées identiques et deux fonctions de transferts de même type mais paramétrées différemment. Lors de l’optimisation, même si PEST arrive à trouver un jeu de paramètres relativement intéressant avec un coefficient de Nash-Sutcliffe $NS = 0.927$ (le coefficient est redéfini si besoin au paragraphe IV.3), les paramètres des deux fonctions de transfert sont identiques (**Tableau II-3**) et on perd donc tout l’intérêt d’avoir deux branches censées représenter deux réservoirs indépendants en parallèle, chacun pouvant avoir sa propre cinétique de vidange.

Tableau II-3 : Valeurs réelles des paramètres et estimation effectuée par PEST

	Paramètre	Valeur réelle	Estimation PEST
Branche 1	gain	0.01	0.355838
	paramètre 1 loi normale	50	-24.9425
	paramètre 2 loi normale	100	26.1651
Branche 2	gain	0.25	0.355838
	paramètre 1 loi normale	0.01	-24.9425

Il est néanmoins possible de résoudre ces deux problèmes en éditant manuellement le fichier de configuration créé par MASH pour PEST. Pour le problème concernant le fait que la loi Gamma n'est pas de classe C^1 sur $]0, +\infty[$, il est possible de modifier les bornes du paramètre n pour avoir une fonction de transfert continûment dérivable. Pour le second problème, relatif au fait que des fonctions en parallèle seront toujours identiques, une solution est de changer les valeurs initiales afin que les paramètres ne convergent pas vers des valeurs identiques. Toutefois, cette modification manuelle est éloignée de l'esprit d'ergonomie que l'on peut attendre d'un logiciel de ce genre, et surtout ne résout pas systématiquement les problèmes. Et enfin, les temps de calculs de PEST sont relativement longs car, dans l'implémentation actuelle, à chaque fois que PEST lance un modèle MASH, une nouvelle instance MATLAB doit s'ouvrir ce qui ralentit considérablement l'optimisation.

Un autre optimiseur a donc été intégré à MASH afin d'avoir un outil 1- automatisé pour faciliter son utilisation, 2- écrit sous MATLAB, afin d'accélérer les calculs 3- basés sur un algorithme différent, permettant de comparer son efficacité et 4- pouvant gérer les problèmes non dérivables.

II.4.2 PSO (Particle Swarm Optimization)

L'algorithme de type optimisation par essaim de particules ou PSO (« Particle Swarm Optimization ») intégré à MASH remplit les quatre critères définis ci-dessus. Son fonctionnement est totalement automatique et son code est écrit directement sous MATLAB comme le reste de MASH afin d'optimiser le temps de calcul. De plus, il fournit un point de comparaison intéressant avec PEST car il utilise un mécanisme d'optimisation complètement différent, étant notamment indifférent au fait que les fonctions de transfert soient continûment dérivables ou non. Et comme cela est expliqué au paragraphe précédent, d'après le théorème « No Free Lunch », il n'y a pas de métaheuristique qui soit meilleure que toutes les autres pour tous les types de problèmes. Le choix s'est donc tourné vers le PSO car il a déjà été utilisé avec succès pour d'autres modèles systémiques en hydrogéologie et que des améliorations intéressantes de l'algorithme ont récemment vu le jour [Chau, 2006; Jiang *et al.*, 2010; Fernández-Martínez *et al.*, 2012; Liu *et al.*, 2013].

De nombreux êtres vivants, en particulier les insectes, sont capables lorsqu'ils sont organisés en société (e.g. en colonie, en essaim, etc.) d'accomplir des tâches beaucoup plus complexes que ce que pourrait accomplir un individu seul. Inspirés par ce phénomène, les métaheuristicques à population, dont fait partie le PSO, sont basées sur une population composée d'individus, généralement appelés « agents » ou « particules » pour le PSO, et où chaque membre obéit à un ensemble de règles relativement simple. Du fait des interactions entre les agents au sein de la population, des comportements émergents apparaissent, dont la complexité est bien supérieure à celle des règles suivies par chaque individu. Avant qu'il ne devienne une métaheuristique d'optimisation, le PSO était initialement conçu pour modéliser le comportement de vols d'oiseaux comme cela est expliqué dans un des articles pionniers qui

introduit cette méthode [Kennedy and Eberhart, 1995]. Afin de recréer le mouvement des vols d'oiseaux, capables de se déplacer et de changer de direction de manière synchronisée, [Heppner and Grenander, 1990] proposent une simulation où chaque agent (représentant un oiseau) possède, à chaque itération, un vecteur vitesse égal à celui de l'agent le plus proche. Le vol a alors un comportement uniforme et tous les agents se déplacent de manière uniforme et sans changer de direction. Pour rendre la simulation plus « vivante », une variable stochastique appelée « folie » est ajoutée. Ainsi à chaque itération, une variation est appliquée à certains vecteurs vitesse choisis au hasard. Ensuite, un vecteur représentant un attracteur (e.g., champ de maïs) est ajouté à la simulation. Ce vecteur étant l'objectif à atteindre de l'essaim, chaque agent est capable d'évaluer sa distance par rapport aux coordonnées de celui-ci. Un des points fondamentaux est que chaque agent est alors capable de se souvenir de la position où il a expérimenté la distance la plus faible entre lui-même et la position objectif : $pbest$. Son vecteur vitesse est alors ajusté afin qu'il se déplace dans le sens qui lui permet de se rapprocher de l'attracteur : un paramètre du système multiplié par une variable aléatoire est ajouté ou retranché au vecteur vitesse actuel. L'autre point fondamental est que les agents sont capables de communiquer les uns avec les autres. Ainsi, chaque agent connaît non seulement la meilleure position qu'il a lui-même expérimenté, mais également la meilleure position que l'essaim tout entier a trouvée : $gbest$. Les vecteurs vitesse sont ainsi mises à jour de la même manière que pour $pbest$: on ajoute ou retranche un paramètre multiplié par une variable aléatoire aux vecteurs vitesse de l'itération en cours afin que l'agent se déplace vers la position $gbest$. Grâce à ces règles, le vol converge de manière très efficace vers l'attracteur. Mais pour passer d'une simulation de vol d'oiseau (problème linéaire en 2-D) à un algorithme d'optimisation (pouvant résoudre des problèmes non-linéaires multidimensionnels) il était nécessaire d'effectuer certaines modifications. Tout d'abord, le terme « folie » fut supprimé, ainsi que la synchronisation des vecteurs vitesse entre les agents les plus proches, le vol (« flock ») devenant un essaim (« swarm »). Ensuite la méthode d'ajustement des vecteurs vitesse fut affinée afin que la quantité ajoutée ou retranchée au vecteur vitesse soit fonction de la différence entre la position actuelle de l'agent et la meilleure position ($pbest$ ou $gbest$). Ainsi, plus la distance entre la position actuelle et la meilleure position locale ou globale est grande, plus la norme du vecteur vitesse est modifiée. La particule a alors tendance à rejoindre plus rapidement la position optimale, et surtout à aller légèrement plus loin, du fait de sa vitesse élevée, ce qui permet d'explorer le voisinage de cet optimum. Finalement, la version « classique » de l'algorithme PSO, présentée dans [Robinson and Rahmat-Samii, 2004], est décrite par l'équation de mise à jour des vecteurs vitesse suivante :

$$v_{n,p}^{i+1} = w * v_{n,p}^i + c_1 \text{rand}() * (pbest_{n,p} - x_{n,p}) + c_2 \text{rand}() * (gbest_{n,p} - x_{n,p}) \quad (\text{II.91})$$

où $v_{n,p}^{i+1}$ désigne la vitesse de la particule p à l'itération $i+1$ dans la dimension n , $x_{n,p}$ est la coordonnée de la particule p dans la dimension n et $\text{rand}()$ est une variable aléatoire distribuée selon une loi uniforme entre 0 et 1. Trois paramètres sont à choisir lors de l'optimisation : le « poids inertiel » w , inférieur à 1, qui détermine dans quelle mesure la particule garde une trajectoire non-modifiée par $pbest$ et $gbest$, le « poids cognitif » c_1 , supérieur à 1, qui détermine à quel point la particule est influencée par sa propre mémoire de la meilleure position et le « poids social » c_2 , supérieur à 1, qui lui caractérise l'influence de la meilleure position

découverte par l'essaim. Le paramètre w , comme l'inertie pour les objets physiques, a pour effet de conserver la trajectoire de la particule sans tenir des effets de son environnement. Plus w est grand et plus les particules ont tendance à conserver leur trajectoire, favorisant l'exploration de l'espace des paramètres. C'est pourquoi dans certaines versions du PSO, ce paramètre décroît au fur et à mesure des itérations, garantissant une bonne exploration au début de l'optimisation, puis une convergence des particules de plus en plus fines. Augmenter le paramètre c_1 a pour effet d'augmenter la capacité exploratoire de l'algorithme, chaque particule privilégiant le suivi de son propre chemin. Au contraire, l'augmentation du paramètre c_2 améliore la convergence globale de l'algorithme car les agents se réfèrent plus à l'expérience de l'essaim qu'à leur propre expérience. Il est bien évidemment nécessaire de trouver un compromis entre exploration et exploitation, ces deux propriétés étant exclusives l'une de l'autre. Favoriser l'exploration permet de trouver de meilleures solutions aux problèmes complexes mais ralentit les calculs à cause d'une moins bonne convergence. Au contraire, mettre l'accent sur l'exploitation permet à l'algorithme de converger rapidement et donc accélère le processus d'optimisation mais peut provoquer des blocages de l'algorithme dans des minima locaux. Le choix des paramètres s'est fait dans un premier temps de manière empirique et des découvertes récentes ont permis de disposer de jeux de paramètres les plus performants possibles comme on le verra ultérieurement.

Après la mise à jour des vecteurs vitesse, l'algorithme procède à la mise à jour des positions :

$$x_{n,p} = v_{n,p} * \Delta t \quad (\text{II.92})$$

où Δt est le pas de temps, généralement égal à 1. Le fonctionnement de l'algorithme PSO de base est expliqué dans les organigrammes de programmation (**Figure II-15** à **Figure II-18**).

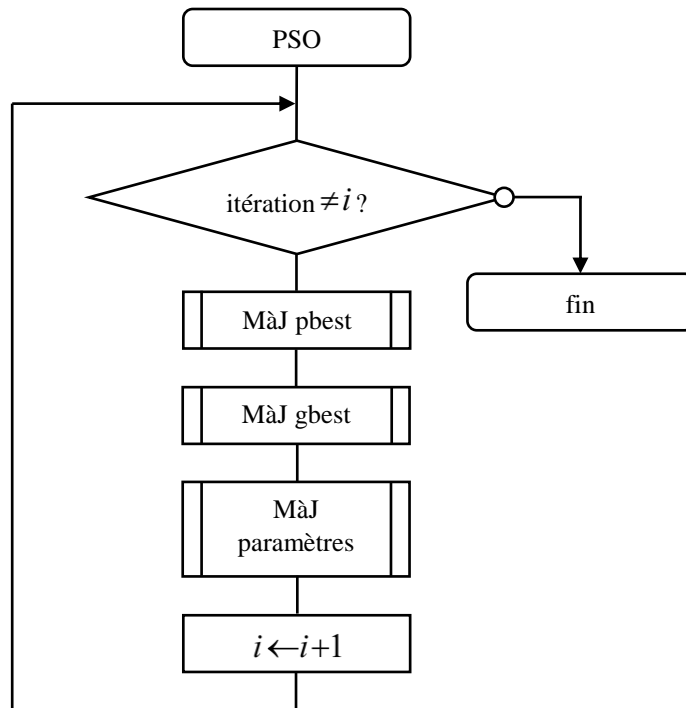


Figure II-15 : Organigramme de programmation de l'algorithme PSO

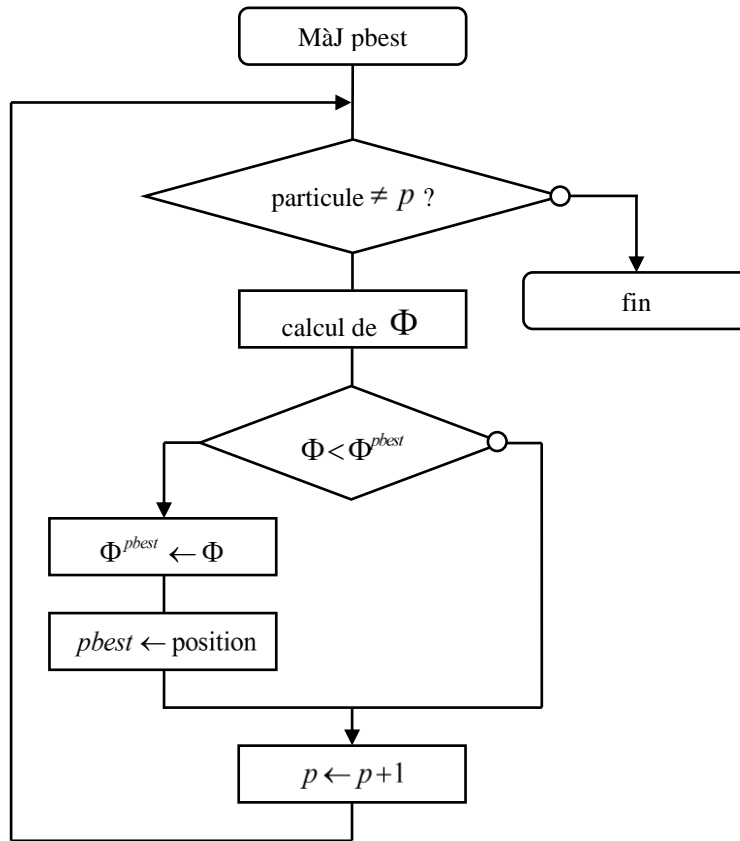


Figure II-16 : Organigramme de programmation du sous-programme « MàJ pbest » chargé de mettre à jour les positions et les valeurs des meilleures fonctions objectifs de chaque particule

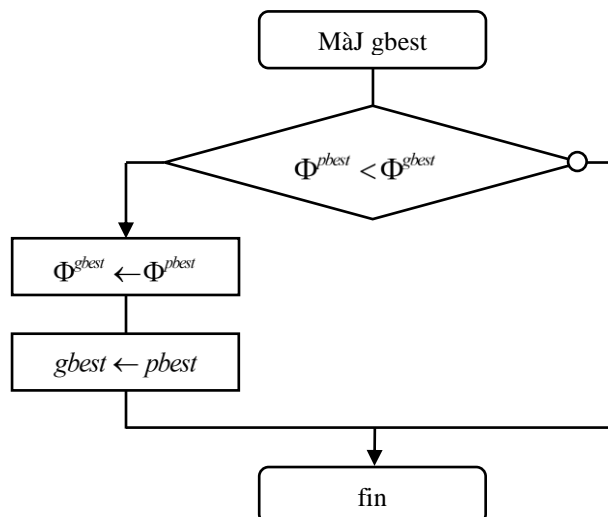


Figure II-17 : Organigramme de programmation du sous-programme « MàJ gbest » chargé de mettre à jour la position et la valeur de la meilleure fonction objectif de l'essaim

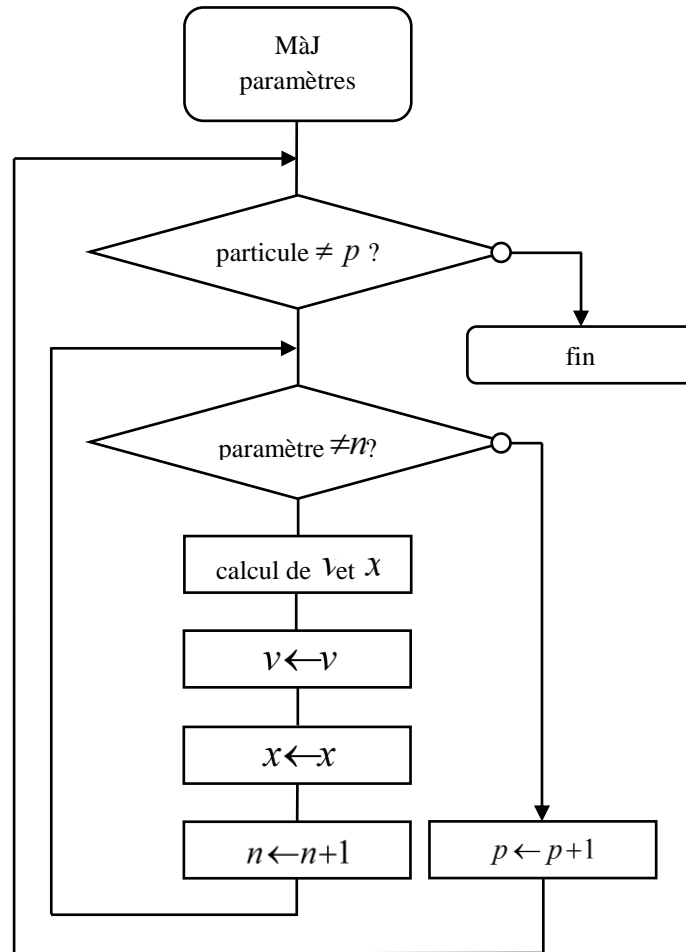


Figure II-18 : Organigramme de programmation du sous-programme « MàJ paramètres » chargé de mettre à jour les vecteurs vitesse et les positions de chaque particule dans chaque dimension

Cette version de l'algorithme est la version « de base » où les différents poids (inertiel, social et cognitif) sont choisis de manière empirique. Récemment, un nouveau développement important a vu le jour. Il s'agit d'une généralisation de l'algorithme que ses auteurs nomment « *PSO family* » [Fernández-Martínez and García-Gonzalo, 2009]. L'idée est d'assimiler l'algorithme PSO, avec ses particules oscillant autour du minimum, à un système poids-ressort oscillant autour de son centre d'attraction par analogie avec la physique, que les auteurs appellent le modèle « PSO continu ». La version de base de l'algorithme qui vient d'être présentée précédemment n'est donc finalement qu'une discrétisation du « PSO continu ». Dès lors, il devient également possible de construire d'autres algorithmes à partir de ce modèle. Selon les centres d'attraction utilisés, les systèmes oscillants réagissent différemment et de même pour les algorithmes, les paramètres de position et de vecteur vitesse des particules peuvent être mis à jour à différents moments lors de l'itération, ce qui induit des comportements plus ou moins exploratoires ou convergents.

Les trois algorithmes de la « *PSO family* » utilisés dans les travaux qui suivent sont :

1. Le « centered-regressive PSO » ou CR-PSO, qui correspond à la version « de base » présentée plus haut. Elle est la plus équilibré entre exploration et convergence. Les positions sont mises à jour à la suite des vecteurs vitesse :

$$\begin{aligned} v(t + \Delta t) &= (1 - (1 - w)\Delta t)v(t) + \Delta t\phi_1(pb_{best}(t) - x(t)) + \Delta t\phi_2(g_{best}(t) - x(t)) \\ x(t + \Delta t) &= x(t) + \Delta tv(t + \Delta t) \end{aligned} \quad (\text{II.93})$$

2. Le « centered-progressive PSO » ou CP-PSO, où la position est mise à jour en même temps que le vecteur vitesse, est la version la plus exploratoire :

$$\begin{aligned} v(t + \Delta t) &= \frac{1 - (\phi_1 + \phi_2)\Delta t^2}{1 + (1 - w)\Delta t} v(t) \\ &\quad + \frac{\Delta t\phi_1}{1 + (1 - w)\Delta t} (pb_{best}(t) - x(t)) \\ &\quad + \frac{\Delta t\phi_2}{1 + (1 - w)\Delta t} (g_{best}(t) - x(t)) \\ x(t + \Delta t) &= x(t) + v(t)\Delta t \end{aligned} \quad (\text{II.94})$$

3. Le « centered-centered PSO » ou CC-PSO, où la position est mise à jour avant le vecteur vitesse en fonction de deux centres d'oscillations consécutifs. Cette forme est la plus convergente :

$$\begin{aligned} x(t + \Delta t) &= x(t) + \left[\begin{aligned} &\frac{2 + (w - 1)\Delta t}{2} v(t) + \frac{\phi_1\Delta t}{2} (pb_{best}(t) - x(t)) \\ &+ \frac{\phi_2\Delta t}{2} (g_{best}(t) - x(t)) \end{aligned} \right] \Delta t \\ v(t + \Delta t) &= \frac{2 + (w - 1)\Delta t}{2 + (1 - w)\Delta t} v(t) \\ &\quad + \sum_{k=0}^1 \left[\begin{aligned} &\frac{\phi_1\Delta t}{2 + (1 - w)\Delta t} (pb_{best}(t + \Delta t) - x(t + \Delta t)) \\ &+ \frac{\phi_2\Delta t}{2 + (1 - w)\Delta t} (g_{best}(t + \Delta t) - x(t + \Delta t)) \end{aligned} \right] \end{aligned} \quad (\text{II.95})$$

Les paramètres de ces trois formules sont les mêmes que ceux de l'équation (II.91), avec en plus ϕ_1 et ϕ_2 qui représentent respectivement les produits entre le poids cognitif c_1 et le poids social c_2 , et une variable aléatoire. Les algorithmes ainsi que toute la théorie de la « *PSO family* » sont décrits plus en détail dans les travaux de [Fernández-Martínez and García-Gonzalo, 2009, 2012]. Un des avantages de cette théorie est qu'elle a permis de définir des zones de stabilité pour les paramètres des algorithmes PSO de manière théorique et non empirique. Tous les paramètres utilisés dans les algorithmes d'optimisation du modèle MASH (i.e. les poids inertiels, sociaux et cognitifs) sont donc situés dans ces zones de stabilité, leur assurant un comportement optimal.

Contrairement au couplage MASH-PEST, les algorithmes PSO peuvent trouver des paramètres différents pour un modèle construit avec des entrées et des fonctions de transfert identiques (voir le problème évoqué au chapitre précédent, et notamment le **Tableau II-3**). En reprenant l'expérience menée précédemment avec PEST, l'algorithme PSO arrive non seulement à trouver un jeu de paramètres excellent (NS = 1) mais ces paramètres correspondent quasi parfaitement aux paramètres réels (**Tableau II-4**).

Tableau II-4 : Valeurs réelles des paramètres et estimation effectuée par le CR-PSO

	Paramètre	Valeur réelle	Estimation CR-PSO
Branche 1	gain	0.01	0.0099963
	paramètre 1 loi normale	50	49.671
	paramètre 2 loi normale	100	101
Branche 2	gain	0.25	0.24987
	paramètre 1 loi normale	0.01	0.015949
	paramètre 2 loi normale	10	9.9938

CHAPITRE III : SIMULATION DE DONNÉES PROVENANT D'UN MODÈLE À BASE PHYSIQUE : LE MODÈLE MARTHE

Ce premier exemple d'application de MASH porte sur un cas d'école synthétique, établi à l'aide d'un modèle à base physique recréant un aquifère simplifié composé d'une à quatre zones ayant des caractéristiques hydrodynamiques différentes. L'objectif de ce test est de vérifier que MASH est capable de reproduire le comportement de systèmes de complexité croissante et de mettre à l'épreuve sa capacité à identifier et à séparer les contributions au débit provenant de différentes zones à l'aide de fonctions de transfert disposées en parallèle.

III.1 Description du modèle à base physique

Afin de disposer d'un cas synthétique dont tous les paramètres sont maîtrisés, plusieurs simulations sont réalisées à l'aide d'un modèle à base physique. Le modèle utilisé est le code MARTHE développé par le BRGM. Ce dernier permet le calcul des écoulements de fluides et des transferts de masse et d'énergie en milieux poreux bidimensionnels et tridimensionnels, via la méthode des différences finies [Thiéry, 2006].

Le modèle créé pour cet exemple est un modèle 2D, composé de mailles carrées. Chaque maille est connectée hydrauliquement aux quatre mailles voisines (nord, sud, est, ouest). Dans chacune de ces mailles, un certain nombre de paramètres sont définis par l'utilisateur (comme les paramètres hydrodynamiques et les conditions aux limites) tandis que d'autres sont calculés par le modèle (comme la charge hydraulique ou le débit dans un tronçon de rivière).

III.1.1 Calcul des charges des nappes

Pour une maille, le calcul des charges en 2D s'écrit de manière implicite comme suit :

$$\sum_{i=1}^4 [T_i \cdot (H_i - H(t))] + Q + R \cdot A = S \cdot A \cdot \frac{H(t) - H(t - dt)}{dt} \quad \text{IV-1}$$

avec T_i la transmissivité entre la maille de calcul et sa voisine i , $H(t)$ la charge de la maille de calcul à la date t , H_i la charge d'une maille voisine, Q le débit prélevé ou injecté, R le flux de recharge, A la surface de la maille et S le coefficient d'emmagasinement.

La charge au centre de la maille de calcul est donc :

$$H(t) = \frac{\sum_{i=1}^4 [T_i \cdot H_i] + Q + R \cdot A + \frac{S \cdot A \cdot H(t - dt)}{dt}}{\sum_{i=1}^4 [T_i] + \frac{S \cdot A}{dt}} \quad \text{IV-2}$$

La résolution s'effectue ensuite par la méthode des gradients conjugués, voir [Thiéry, 2004]. Le terme de recharge pour une maille donnée (ainsi que le terme de ruissellement évoqué par la suite pour le calcul des débits de rivières) est calculé grâce à un bilan hydro-climatique. Ce calcul est effectué par le code GARDÉNIA (*modèle Global A Réservoirs pour la simulation des DÉbits et des Niveaux Aquifères*) [Thiéry, 2003, 2012a], un modèle conceptuel selon la terminologie du chapitre II intégré à MARTHE. Dans chaque zone hydro-climatique (appelée aussi « zone de sol »), le module GARDÉNIA utilise des données météorologiques (pluie et ETP) ainsi que des paramètres censés décrire le sol de la zone en question (réserve disponible pour l'évapotranspiration (RDE), durée de demi-percolation (DDP) et niveau d'équilibre ruissellement/percolation (NERP)) pour calculer la répartition entre écoulement rapide (ruissellement) et écoulement lent (percolation vers la nappe). Dans leur rapport sur le modèle des aquifères du Crétacé en Charentes, [Douez and Bichot, 2012] décrivent le fonctionnement des deux réservoirs de GARDÉNIA (**Figure III-1**) :

- Le premier réservoir modélise la partie supérieure du sol, sa hauteur maximale étant fixée par la RDE. Il se vide ou se remplit suivant la pluie et l'ETP. L'alimentation du réservoir sous-jacent est possible lorsque la capacité de rétention du sol est atteinte.
- Le second réservoir modélise la zone non saturée au-dessus du niveau de nappe. Il est alimenté par le premier réservoir et est vidangé par percolation dans l'aquifère suivant une loi exponentielle de constante de temps DDP et par ruissellement suivant le paramètre NERP.

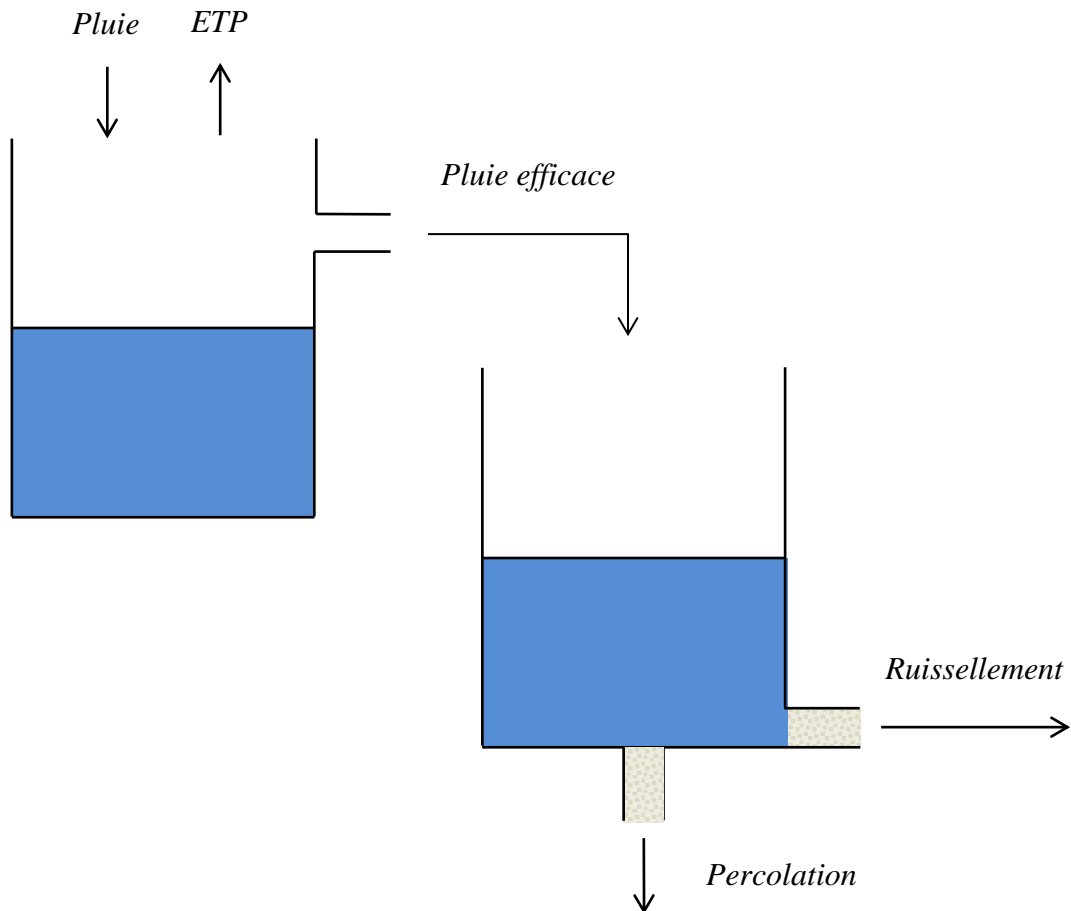


Figure III-1 : Schéma de fonctionnement des réservoirs du logiciel GARDÉNIA (adapté d'après [Thiéry, 2003])

III.1.2 Calcul des débits des rivières

Le réseau hydrographique est géré par MARTHE sous la forme d'un système de tronçons de rivières organisés, superposé au maillage [Thiéry, 2012b]. Chaque tronçon est décrit par sa géométrie (longueur, largeur, et altitude du lit), par ses propriétés physiques (rugosité du lit, pente du lit, épaisseur et perméabilité du lit et des berges) et par son état (hauteur d'eau dans la rivière). Pour le modèle simple construit ici, le bilan des débits dans un tronçon de rivière s'écrit :

$$Q_{aval} = Q_{amont} + Q_{éch.} + Q_{ruis.} \quad \text{IV-3}$$

où Q_{aval} est le débit aval, Q_{amont} le débit amont, $Q_{éch.}$ le débit échangé avec la nappe et $Q_{ruis.}$ le débit provenant du ruissellement. La hauteur d'eau quant à elle est reliée au débit aval par la formule de Manning-Strickler. Grâce à ces équations, le débit dans l'ensemble du réseau hydrographique est obtenu par un processus itératif, en utilisant l'ordre de Strahler de chaque tronçon. Un débit peut être échangé depuis un tronçon de rivière vers une nappe, et réciproquement depuis une nappe vers la rivière. Trois cas peuvent se produire :

- Si le niveau dans la nappe est inférieur au lit de la rivière, le débit $Q_{perc.}$ est un terme de percolation, indépendant du niveau de la nappe :

$$Q_{perc.} = \frac{A_{éch.} \cdot K}{e} \cdot (H_{riv.} - H_{lit}) \quad \text{IV-4}$$

avec $A_{éch.}$ la surface d'échange, K la perméabilité du lit et des berges, e l'épaisseur du lit et des berges, $H_{riv.}$ la cote de la surface libre de la rivière et H_{lit} la cote du lit de la rivière.

- Si le niveau dans la nappe est supérieur au lit de la rivière, et inférieur à la cote du sol, le débit $Q_{éch.}$ est proportionnel à la différence entre la hauteur d'eau dans la rivière et le niveau dans la nappe :

$$Q_{éch.} = \frac{A_{éch.} \cdot K}{e} \cdot (H_{riv.} - H_{nappe}) \quad \text{IV-5}$$

avec H_{nappe} la charge hydraulique de la nappe.

- Le niveau dans la nappe est supérieur à la cote du sol : le débit d'échange est augmenté d'un terme complémentaire de débordement de la nappe appliqué à la surface de la maille :

$$Q_{débo.} = \frac{(A - A_{tronçon}) \cdot K}{e} \cdot (H_{nappe} - H_{topo}) \quad \text{IV-6}$$

avec A la surface de la maille, $A_{tronçon}$ la surface du tronçon et H_{topo} la cote de la surface topographique.

III.2 Définition des paramètres de l'expérience

Grâce au modèle MARTHE décrit ci-dessus, des simulations vont permettre d'obtenir des chroniques fictives de débit à partir de chroniques de pluie et d'ETP, en faisant varier certains paramètres du modèle (**Figure III-2**). Une fois ces chroniques de sorties générées, le but de l'expérience est de voir si MASH est capable de les reproduire, connaissant les chroniques d'entrée.

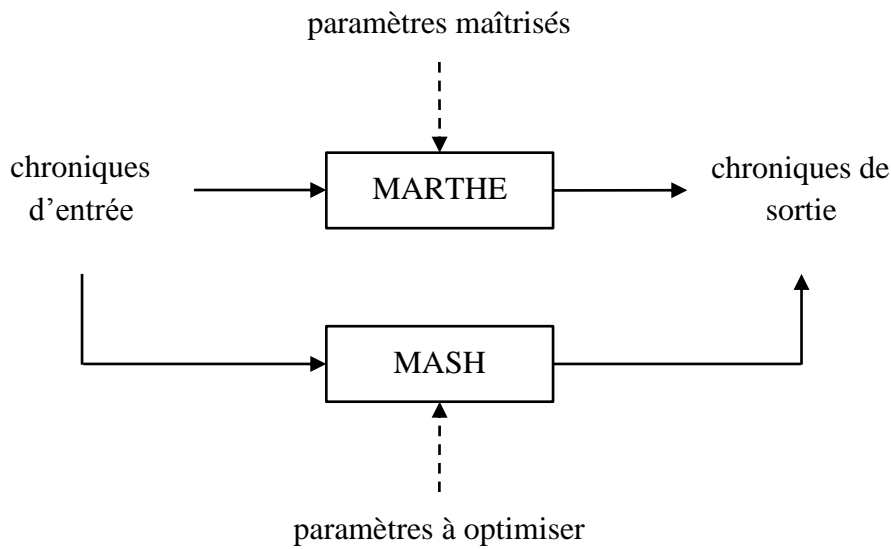


Figure III-2 : Principe de fonctionnement de l'étude : MARTHE simule des chroniques de sortie en fonction des chroniques d'entrée et de paramètres maîtrisés. MASH essaie ensuite de reproduire les chroniques de sorties ainsi créées.

Au cours des différentes simulations, le modèle MARTHE simule l'écoulement 2D d'une nappe libre drainée par une rivière. Le but d'une telle approche est de tester si le modèle MASH est capable d'identifier des zones de la nappe ayant des contributions différentes au débit de la rivière, grâce à son architecture associant des fonctions de transfert réparties selon des branches parallèles. La grille est donc soit uniforme (Figure III-3, a), soit divisée en plusieurs zones de sol possédant chacune des propriétés différentes, (Figure III-3, b, c et d).

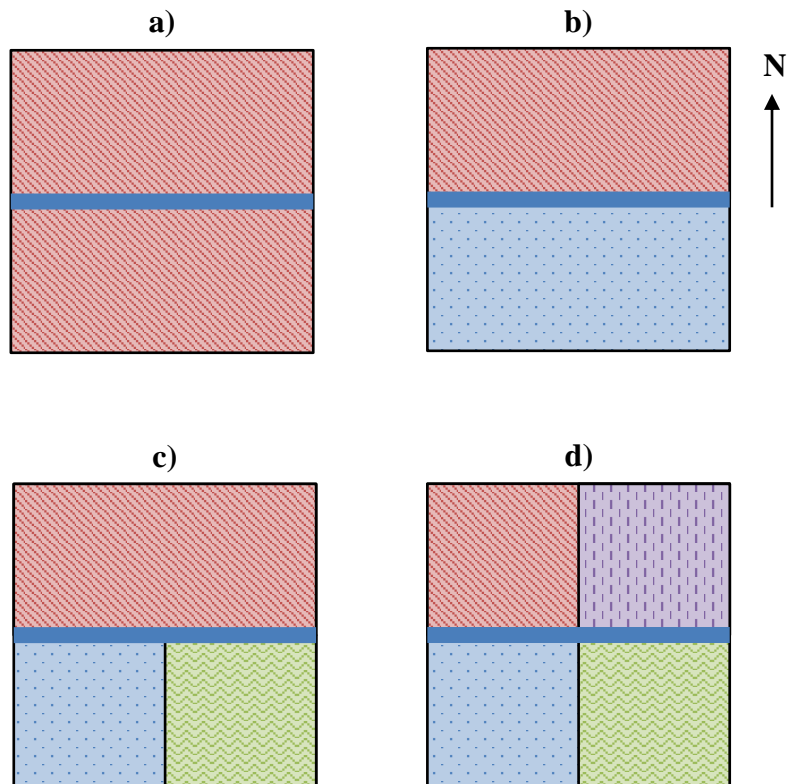


Figure III-3 : Représentation schématique des différents modèles comportant (a) une zone ; (b) deux zones distinctes ; (c) trois zones distinctes ; (d) quatre zones distinctes. Le trait épais horizontal représente la rivière.

Le maillage du cas synthétique est un parallélépipède comportant 10000 mailles (100x100x1) traversé en son milieu par une rivière. Les conditions aux limites sont deux charges imposées : -2m à l'ouest et -3m à l'est. Afin de créer des zones ayant des comportements différents, deux paramètres de MARTHE sont modifiés : la durée de demi-percolation (DDP) et le niveau d'équilibre ruissellement/ percolation (NERP), voir paragraphe III.1.1. Ces paramètres ont l'impact le plus important sur la réponse en débit, du fait qu'ils modifient la quantité d'eau ruisselant vers la rivière ou percolant vers la nappe :

- la DDP modifie la forme de la période de récession – plus sa valeur est élevée, plus la durée de la récession sera grande (**Figure III-4, a**). De la même manière que la durée de demi-vie d'un élément radioactif renseigne sur le temps nécessaire à désintégrer les noyaux de cet élément, la DDP indique le temps nécessaire pour vidanger par percolation la zone non saturée vers la nappe.
- le NERP est proportionnel à la réponse en débit du système à la suite d'un évènement pluvieux – plus sa valeur est élevée, moins la réponse en débit est forte (**Figure III-4, b**). En effet, plus le NERP est élevé, plus la quantité d'eau parvenant à la rivière par ruissellement est faible, ce qui a pour effet de diminuer la réponse en débit.

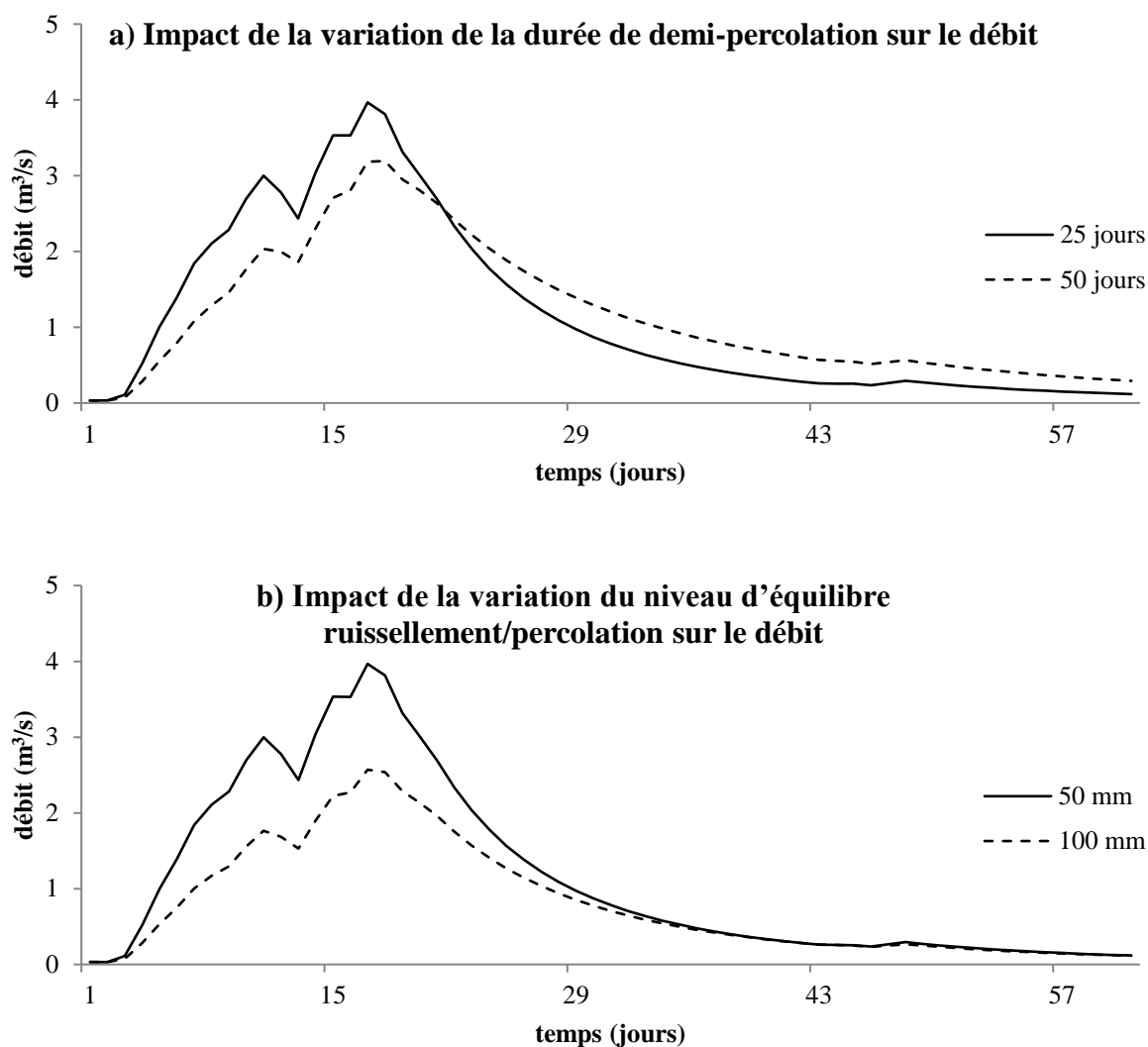


Figure III-4 : Évolution du débit en fonction du temps, suite à une série d'évènements pluvieux, calculé avec le modèle à base physique MARTHE en faisant varier (a) la durée de demi-percolation à 25 jours et à 50 jours (NERP = 50 mm) ; (b) le niveau d'équilibre ruissellement/percolation à 50 mm et 100 mm (DDP = 25 jours)

Lors des simulations, quatre jeux de paramètres, correspondant à quatre natures de terrain différentes, sont affectés successivement aux zones de sol du modèle (**Tableau III-1**). Ces paramètres sont à la fois suffisamment différents pour que les réponses en débit soient contrastées, et suffisamment proches pour tester la sensibilité de MASH dans l'identification de zones peu différenciées.

Tableau III-1 : Jeux de paramètres utilisés pour les différentes zones de l'exemple

Nom du jeu de paramètres	Durée de demi-percolation (en jours)	Niveau d'équilibre ruissellement/percolation (en mm)
A	25	50
B	50	100

C	100	200
D	200	400

Le modèle à base physique MARTHE reçoit en entrée deux chroniques : la pluie brute et l'ETP. Ces deux séries de données sont au pas de temps journalier (1502 mesures, du 08/07/1993 au 17/08/1997) et sont extraites de chroniques réelles. À partir de ces chroniques, MARTHE calcule en écoulement transitoire le débit de la rivière pour ces 1502 pas de temps. Une fois que l'on dispose de cette chronique de débit simulée, MASH utilise en entrée les mêmes chroniques de pluie et d'ETP pour tenter de reproduire cette chronique de débit. Les paramètres des fonctions de transfert sont alors optimisés grâce à l'algorithme PSO défini au chapitre précédent. Pour cette série de test, le PSO est lancé 3 fois avec 50 particules durant 200 itérations et seule la meilleure solution (i.e. la meilleure particule au cours de l'ensemble des itérations) est conservée. C'est avec cette solution que sont calculés les débits de sortie.

Dans chacun des essais, le modèle MASH possède autant de branches parallèles que le modèle MARTHE possède de zones ayant des propriétés différentes. Au sein de ces différentes branches, la loi Gamma est la seule fonction de transfert utilisée car c'est celle qui donne les meilleurs résultats pour les simulations à une zone. Cela s'explique sans doute par le fait que l'écoulement est macroscopiquement diffusif, et que la loi Gamma, étant une solution générique de diffusion radiale, permet de bien mimer le système.

III.3 Reproduction du débit à l'exutoire

La première série de simulation est très simple : on applique successivement les jeux de paramètres A à D (**Tableau III-1**) à toute la grille du modèle MARTHE (**Figure III-3, a**). On tente ensuite de reproduire les débits simulés par MARTHE avec le modèle MASH. Comme le modèle MARTHE ne comporte qu'une zone, le modèle MASH ne comporte qu'une branche. L'architecture MASH est donc très simple : une entrée, une sortie et une fonction de transfert, soit trois paramètres à optimiser (le gain de la branche et les deux paramètres de la fonction de transfert en loi Gamma).

III.3.1 Modèles à une zone

Sur la **Figure III-5** sont tracées les chroniques de débits simulées par MARTHE et reproduites par MASH.

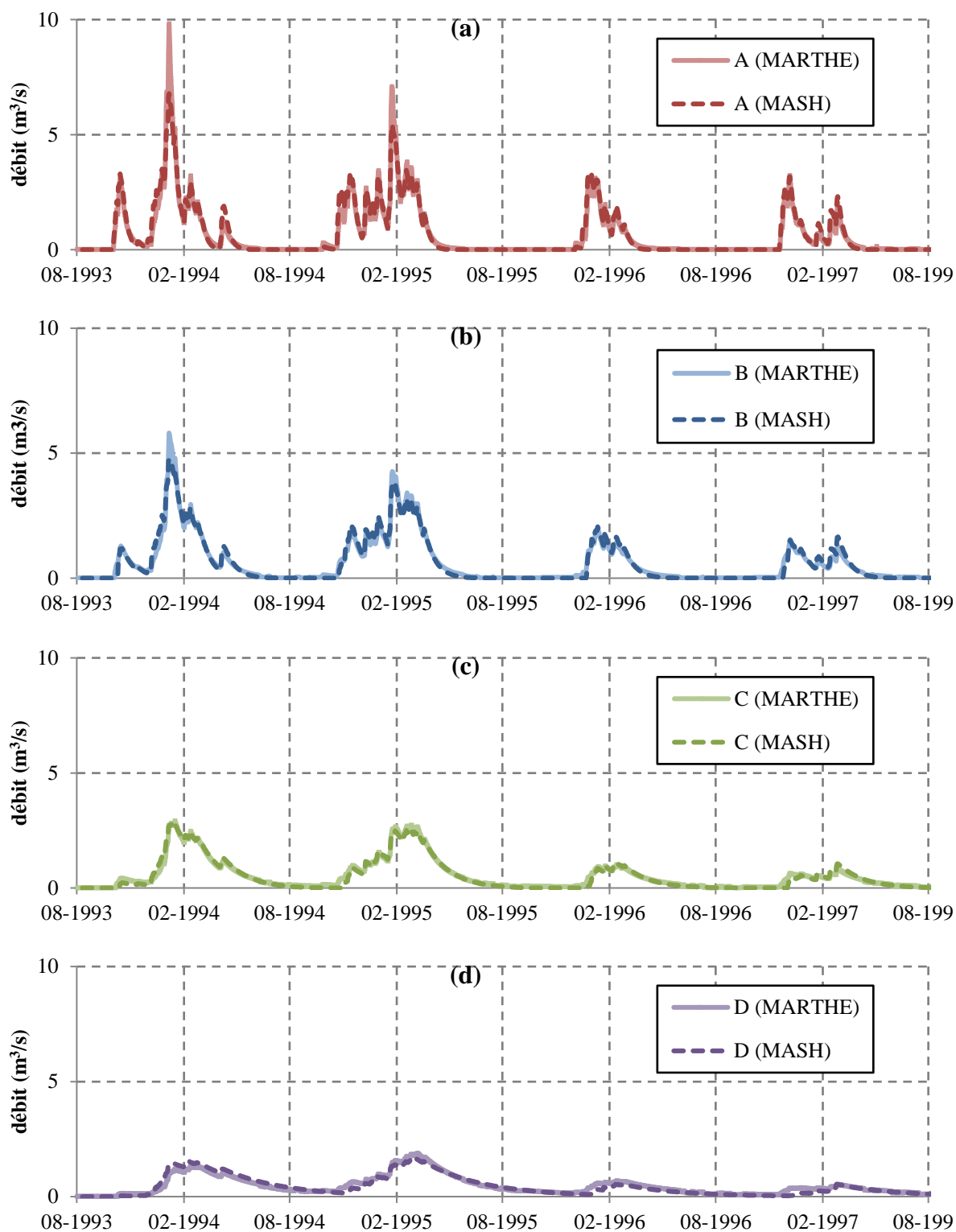


Figure III-5 : Comparaison des débits simulés par MARTHE et recréés par MASH pour quatre types de comportement d'un bassin versant composé d'une seule zone (jeux de paramètres A à D du Tableau III-1)

Afin de quantifier la qualité de l'ajustement entre ces deux chroniques, on utilise le coefficient de Nash-Sutcliffe [Nash and Sutcliffe, 1970] défini par :

$$NS = 1 - \frac{\sum_{t=1}^T (Q_{obs}^t - Q_{mod}^t)^2}{\sum_{t=1}^T (Q_{obs}^t - \overline{Q_{obs}})^2} \quad \text{IV-7}$$

où $\overline{Q_{obs}}$ est le débit moyen observé, Q_{obs}^t est le débit journalier observé et Q_{mod}^t est le débit journalier modélisé. NS est compris entre $-\infty$ et 1, et croit proportionnellement à la qualité de l'ajustement. Ce coefficient est le rapport entre l'erreur quadratique moyenne et la variance, soustrait à 1. L'erreur quadratique moyenne mesure les variations entre les données observées et les débits modélisés. La variance mesure les variations entre les données observées par rapport à la moyenne de ces valeurs. Ainsi $NS = 1$ équivaut à un modèle parfait où les données observées et simulées sont égales. $NS = 0$ signifie que le débit observé moyen est un aussi bon prédicteur que le modèle et $NS < 0$ que le débit moyen est un meilleur prédicteur que le modèle [Legates and McCabe, 1999]. Dans chacun des tableaux suivants où sont rapportés des coefficients de Nash-Sutcliffe, un code couleur est attribué afin d'identifier immédiatement la qualité de l'ajustement : dégradé du vert au jaune pour $1 > NS > 0.5$; dégradé du jaune au rouge pour $0.5 \geq NS \geq 0$; rouge pour $NS < 0$. Les coefficients NS mesurant la qualité de l'ajustement entre les débits MARTHE et les débits reproduits par MASH pour le modèle à une zone sont donnés dans le **Tableau III-2**.

Tableau III-2 : Mesure de la qualité de l'ajustement entre les débits simulés par MARTHE et recréés par MASH pour quatre types de comportement du bassin versant

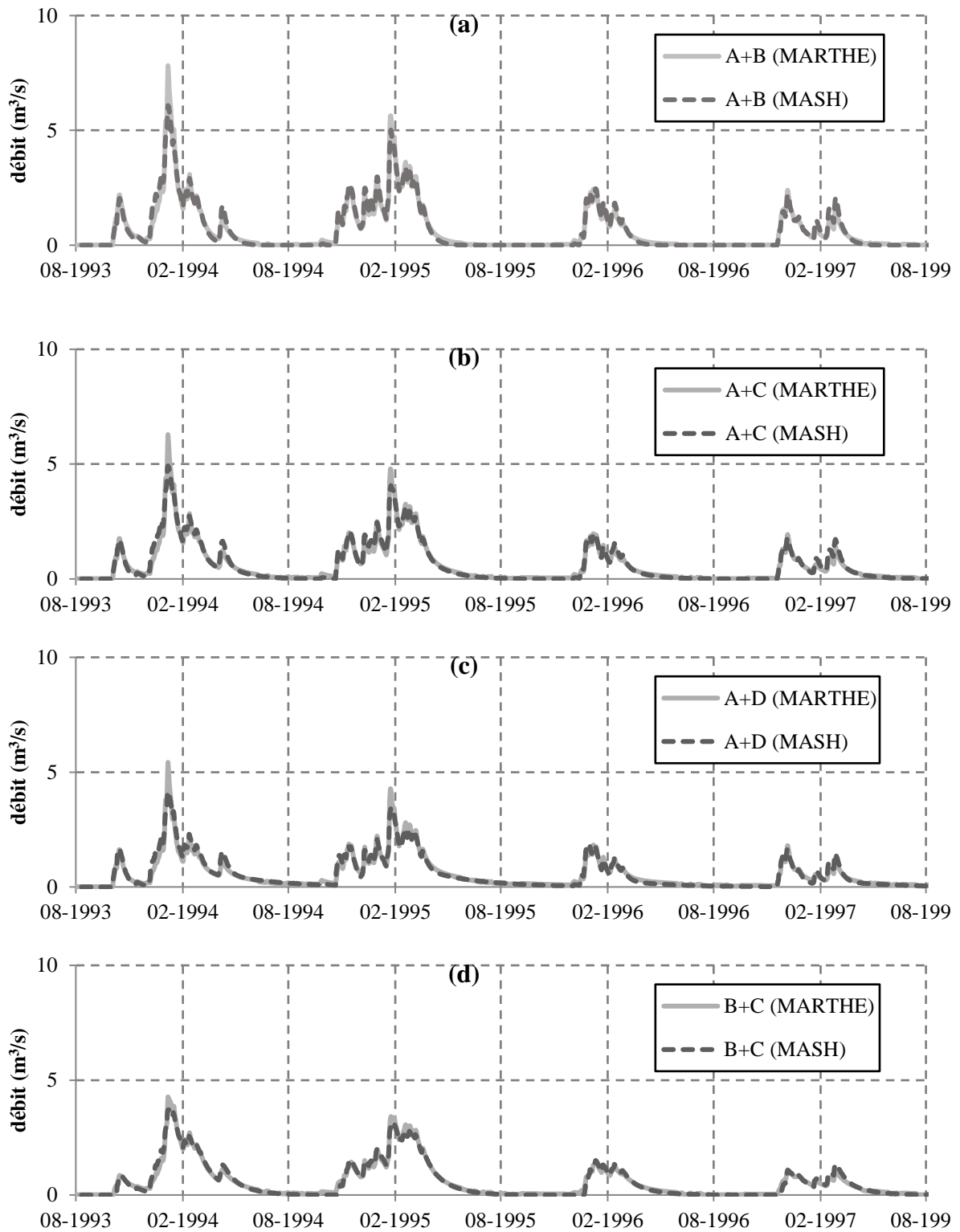
Jeux de paramètres	NS
A	0.959
B	0.978
C	0.976
D	0.901

Comme prévu, au fur et à mesure que la durée de demi-percolation et que le niveau d'équilibre ruissellement/percolation augmente (de A vers D), la réponse en débit devient moins élevée et la période de récession devient plus longue. MASH arrive à très bien reproduire les débits, avec $NS < 0.9$, quels que soit les paramètres du bassin versant. Le jeu de paramètre D, correspondant au période de récession les plus élevées, est légèrement moins bien reproduit que les autres.

III.3.2 Modèles à deux, trois et quatre zones

La même expérience est ensuite reproduite avec un modèle MARTHE divisé successivement en deux, trois et quatre zones (**Figure III-3**, b, c et d). Les modèles MASH associés comprennent autant de branches parallèles que les modèles MARTHE comptent de zones. Les modèles MASH à deux, trois et quatre branches possèdent respectivement six, neuf et douze paramètres,

correspondants à un gain par branche et deux paramètres par fonction de transfert. Les courbes de débits sont tracées dans les **Figure III-6**, **Figure III-7** et **Figure III-8** et les coefficients de Nash Sutcliffe correspondants sont donnés dans les **Tableau III-3**, **Tableau III-4** et **Tableau III-5**.



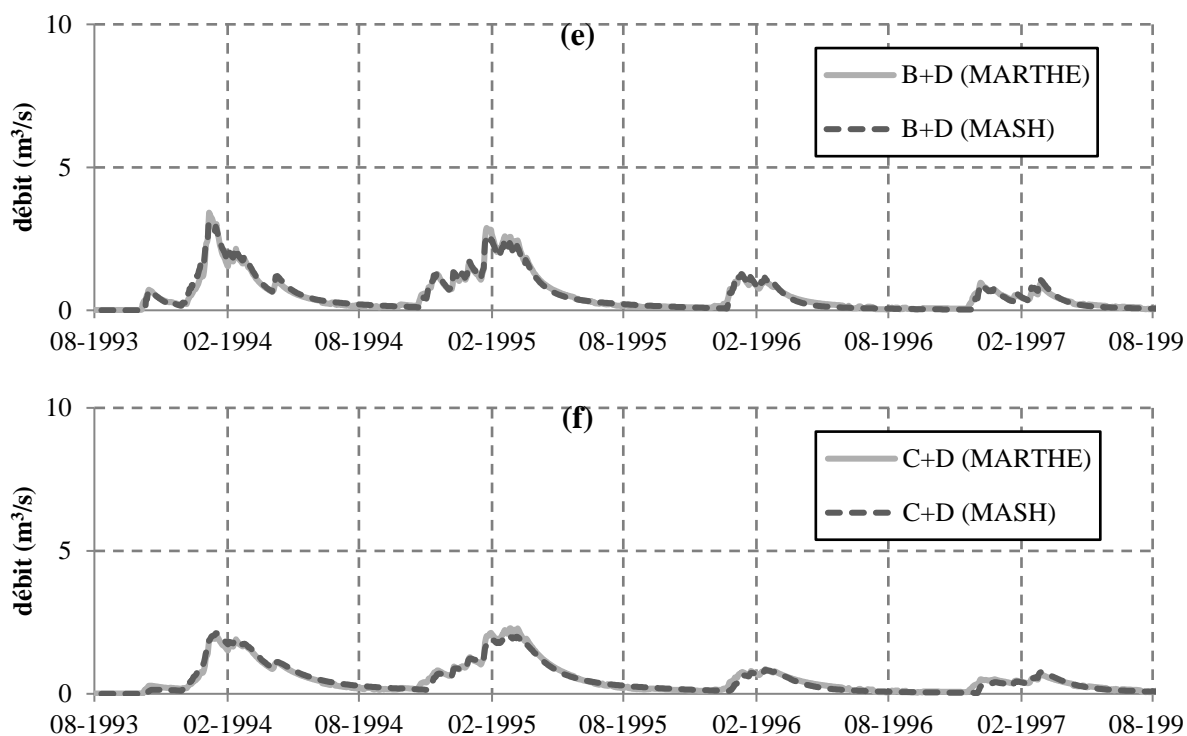


Figure III-6 : Comparaison des débits simulés par MARTHE et recréés par MASH pour six types de comportement du bassin versant divisé en deux zones

Tableau III-3 : Mesure de la qualité de l'ajustement entre des débits simulés par MARTHE et recréés par MASH pour six types de comportement du bassin versant divisé en deux zones

Jeux de paramètres	NS
A+B	0.977
A+C	0.978
A+D	0.972
B+C	0.984
B+D	0.981
C+D	0.971

Le modèle MARTHE divisé en trois zones est asymétrique (Figure III-3, c) : la zone nord est deux fois plus grande que les zones sud-ouest et sud-est. Avec quatre jeux de paramètres il est donc possible de construire douze cas différents mais seuls quatre ont été utilisés : A+B+C, B+D+A, C+D+A et D+B+C où la première lettre désigne le jeu de paramètre appliqué à la plus grande zone.

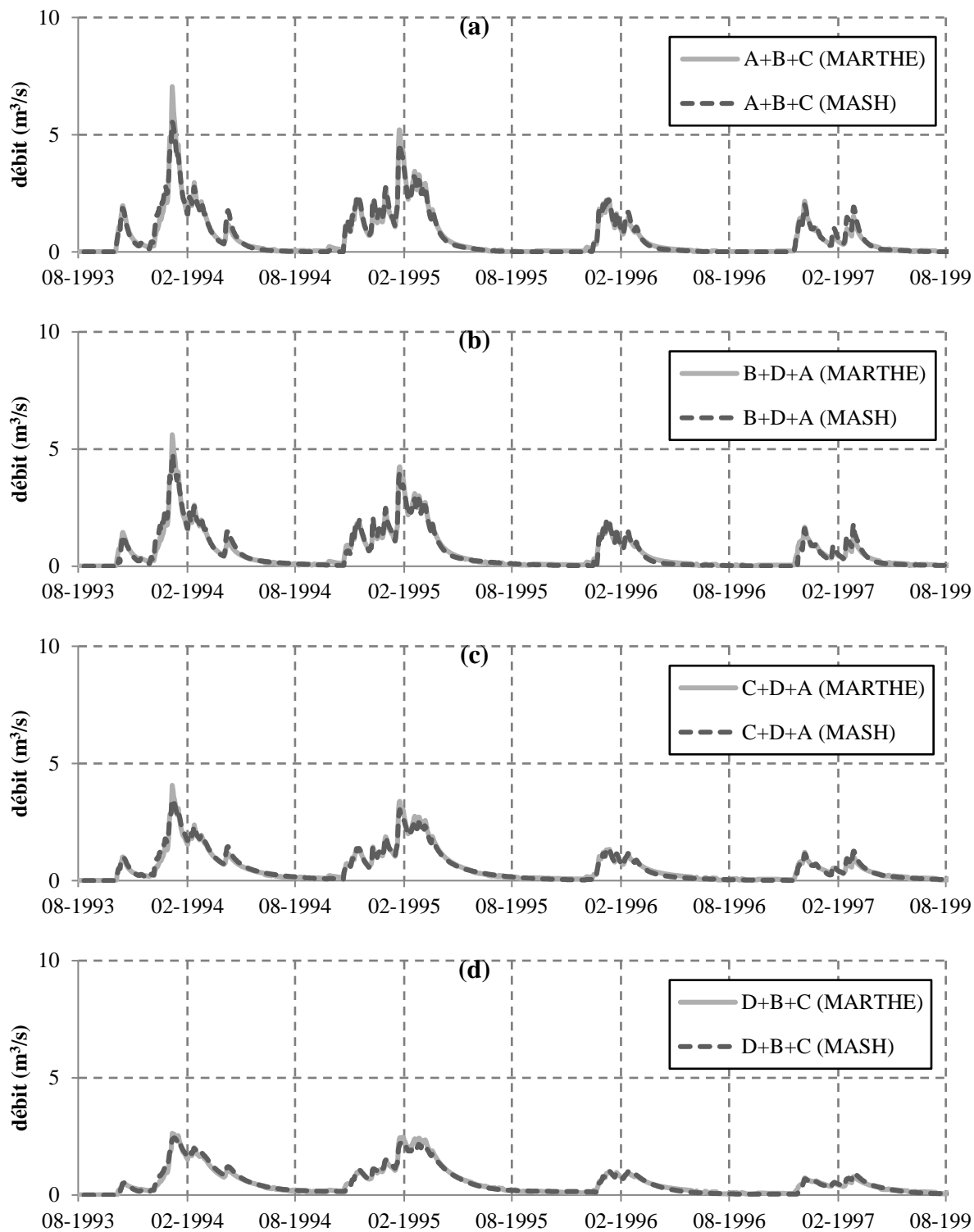


Figure III-7 : Comparaison des débits simulés par MARTHE et recréés par MASH pour quatre types de comportement du bassin versant divisé en trois zones

Tableau III-4 : Mesure de la qualité de l'ajustement entre des débits simulés par MARTHE et recréés par MASH pour quatre types de comportement du bassin versant divisé en trois zones

Jeux de paramètre	NS
A+B+C	0.979
B+D+A	0.977
C+D+A	0.985
D+B+C	0.984

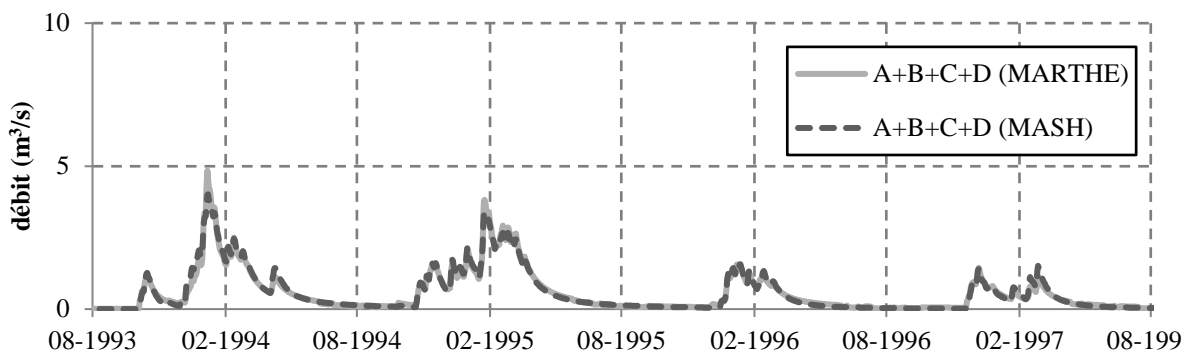


Figure III-8 : Comparaison des débits simulés par MARTHE et recréés par MASH pour le bassin versant divisé en quatre zones

Tableau III-5 : Mesure de la qualité de l'ajustement entre des débits simulés par MARTHE et recréés par MASH pour le bassin versant divisé en quatre zones

Jeux de paramètre	NS
A+B+C+D	0.984

Quel que soit le nombre de zone que possède le modèle MARTHE, le modèle MASH reproduit très bien le débit à l'exutoire du bassin, avec des *NS* systématiquement supérieurs à 0.97. On peut néanmoins se demander si, en plus de réussir à simuler le débit total, MASH est capable d'identifier les contributions provenant de chacune des zones aux caractéristiques différentes, grâce à son architecture parallèle. Cette question est abordée au paragraphe suivant.

III.4 Contribution des différentes zones au débit et fonctions de transfert

Il a été établi au paragraphe précédent que le modèle MASH à une branche était quasiment équivalent au modèle MARTHE représentant un bassin uniforme (modèle à une zone) (Figure III-9). Dans l'idéal, il faudrait donc que les fonctions de transfert optimisées par MASH gardent la même forme lorsqu'elles représentent les mêmes zones de sol, quel que soit le nombre de zones du modèle MARTHE. Et de même, la contribution au débit pour une branche donnée de

MASH devrait être la même, quel que soit le nombre de zone, pourvu que la fonction de transfert de la branche représente la même zone.

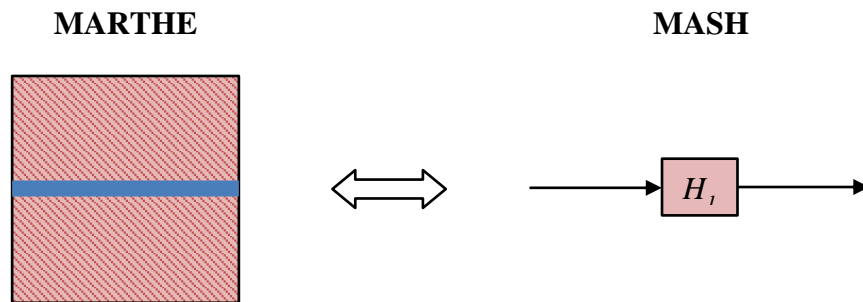


Figure III-9 : Les fonctions de transfert à une branche représentent efficacement les modèles à une zone

La seconde partie de l'étude porte donc sur les contributions au débit total des différentes branches de MASH et sur les fonctions de transfert qui y sont associées. La question est de savoir si les branches des modèles MASH représentent de manière univoque les différentes zones des modèles MARTHE (**Figure III-10**).

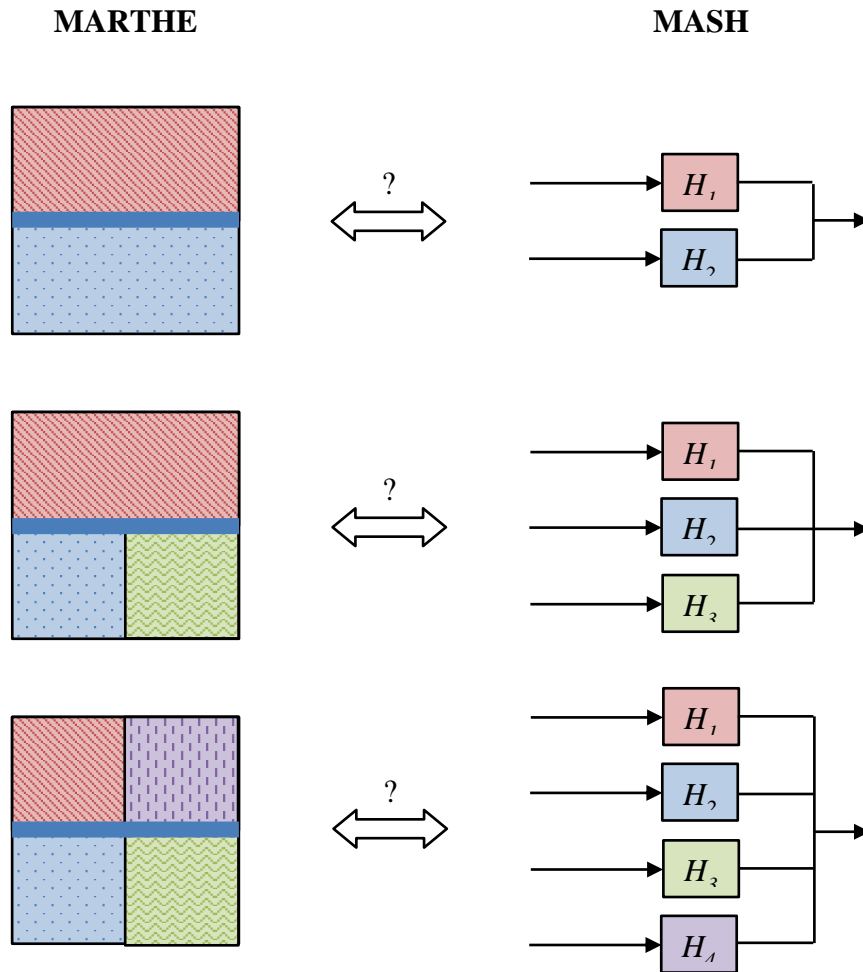


Figure III-10 : Représentation schématique de l'étude : les fonctions de transfert des modèles MASH représentent-elles les zones des modèles MARTHE ?

III.4.1 Modèles à deux branches

Dans un premier temps, les fonctions de transfert du modèle à une branche (correspondant à une zone de sol) sont donc comparées aux fonctions des modèles à deux branches (pour les mêmes zones de sol) (Figure III-11). De la même manière, le débit simulé par MASH pour un modèle à une zone est comparé à la contribution au débit des modèles MASH à deux branches (Figure III-12).

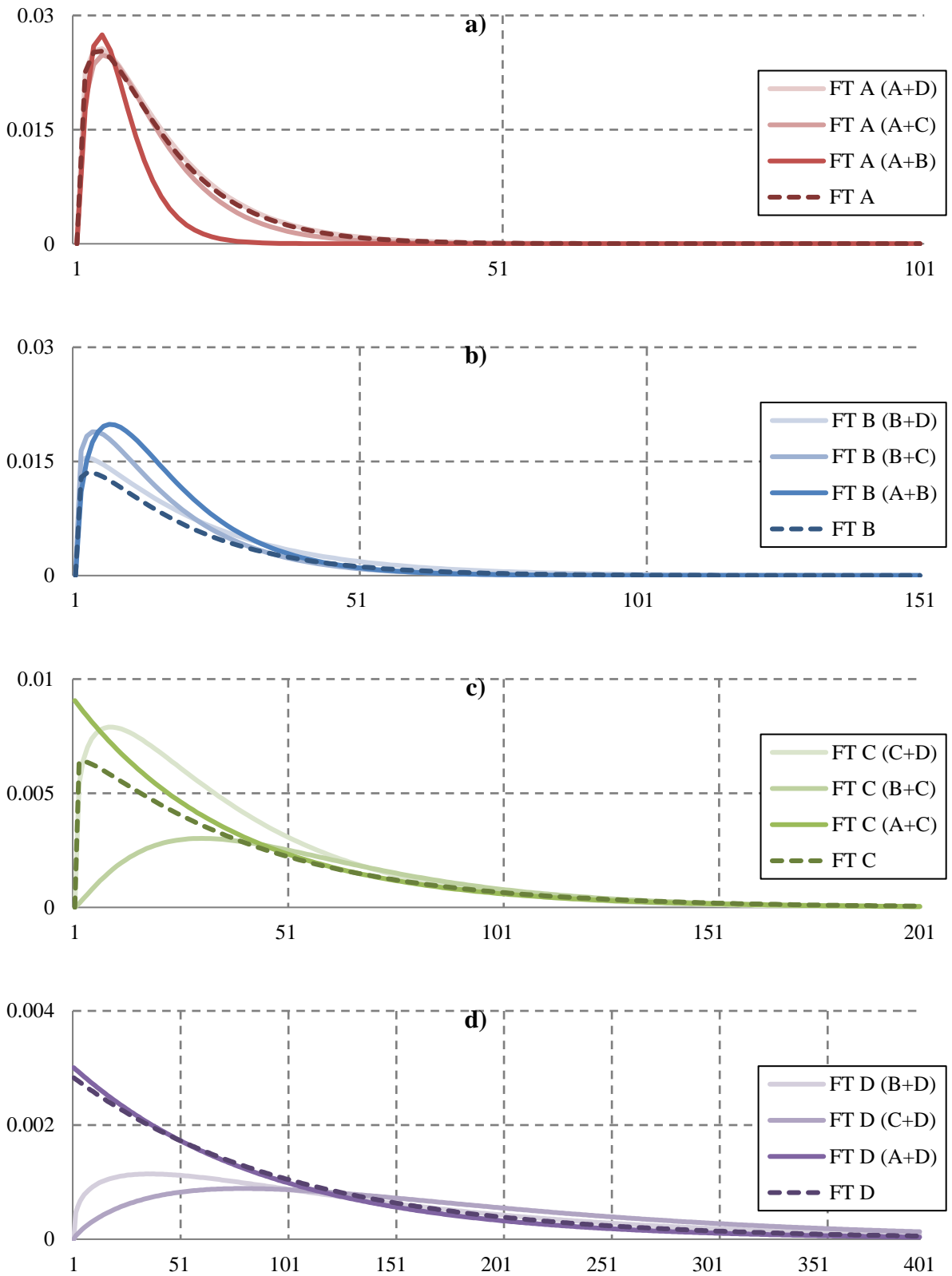


Figure III-11 : Comparaison entre les fonctions de transfert calculées pour un modèle à une zone (en trait pointillé) et celles pour un modèle à deux zones (en trait plein)

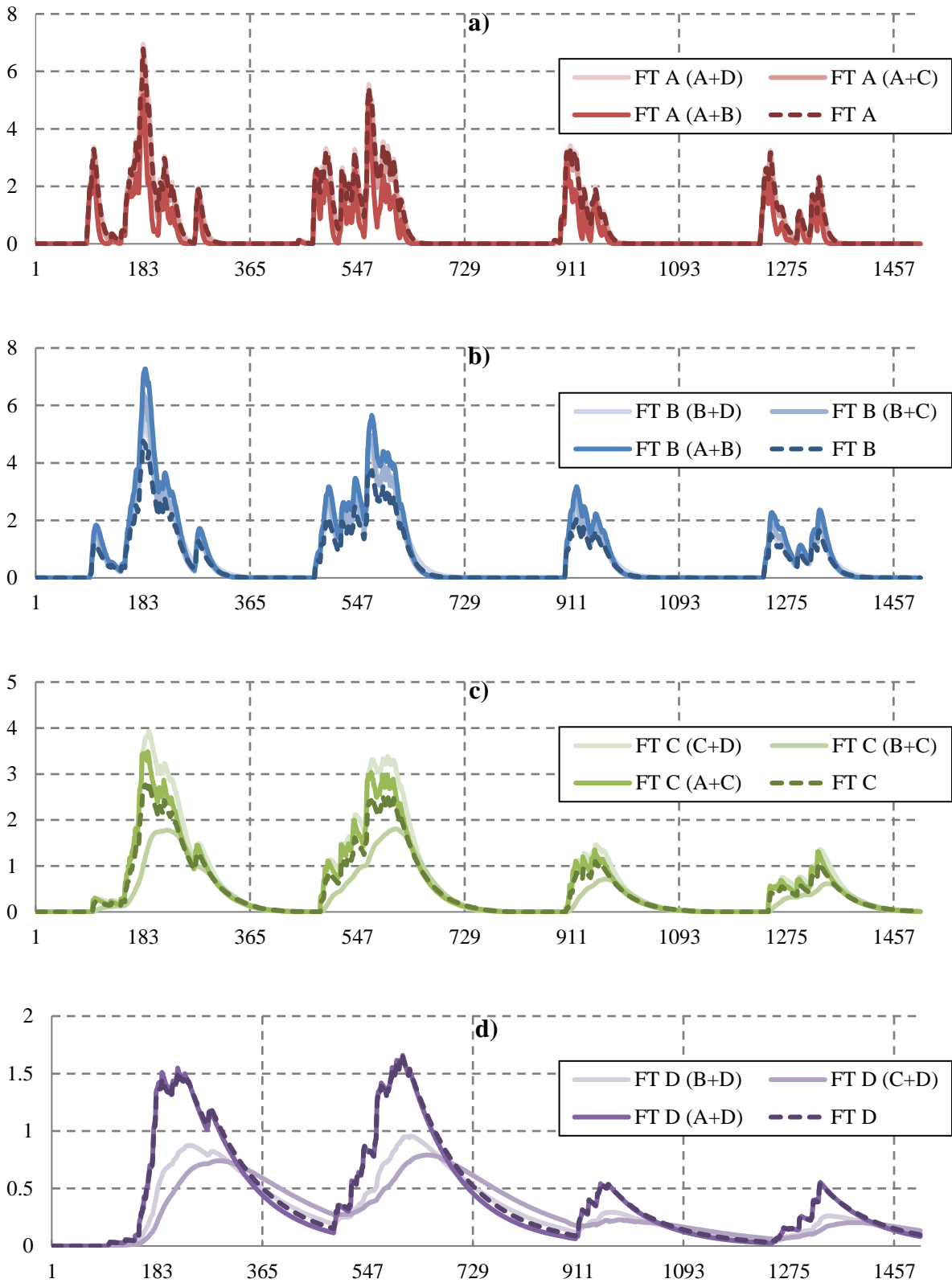


Figure III-12 : Comparaison entre les débits calculés pour un modèle à une zone (en trait pointillé) et ceux pour un modèle à deux zones (en trait plein)

Les coefficients de Nash-Sutcliffe reportés dans le **Tableau III-6** mesurent l'ajustement entre les fonctions de transfert du modèle MASH à une zone (pris comme référence, étant donné le très

bon ajustement entre les modèles MARTHE à un zone et les modèles MASH à une branche) et celles du modèle à deux zones, Même si ces coefficients sont normalement destinés à quantifier l'ajustement entre des chroniques hydrologiques (typiquement des débits), ils possèdent l'avantage d'être indépendants du nombre de données utilisées (équation (IV-7)), permettant ainsi de comparer entre elles des fonctions de transfert de longueurs différentes.

Tableau III-6 : Mesure de la qualité de l'ajustement entre les fonctions de transfert des modèles MASH à une branche et ceux à deux branches

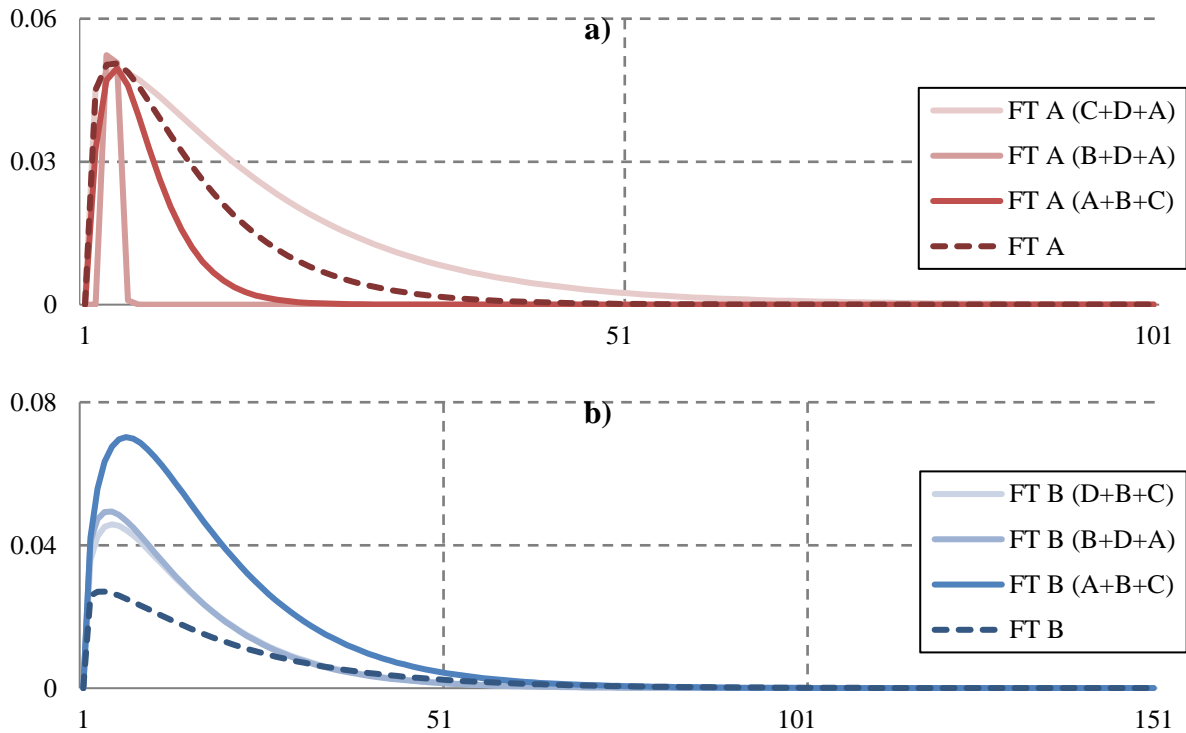
Fonction de transfert	NS	Rapport entre les paramètres des deux zones
FT A (A+D)	0.999	8
FT A (A+C)	0.992	4
FT A (A+B)	0.820	2
FT B (B+D)	0.948	4
FT B (B+C)	0.799	2
FT B (A+B)	0.503	2
FT C (C+D)	0.643	2
FT C (B+C)	0.236	2
FT C (A+C)	0.719	4
FT D (B+D)	0.347	4
FT D (C+D)	-0.216	2
FT D (A+D)	0.991	8

Le modèle A+D composé des zones les plus différenciées (rapport entre les paramètres égal à 8) permet d'obtenir des fonctions de transfert très ressemblantes aux FT du modèle à une zone (NS=0.999 et NS=0.991 respectivement pour la FT A et la FT D). Les modèles composés A+C et B+D de zones moyennement différenciées (rapport entre les paramètres égal à 4) reproduisent correctement les fonctions de transfert, mais moins bien que les modèles possédant des zones plus différenciées (NS=0.992, NS=0.948, NS=0.719 et NS=0.347 respectivement pour les FT A, B, C et D). Les modèles composés des zones les moins différenciées (rapport entre les paramètres égal à 2) sont ceux qui vont le moins bien reproduire les différentes FT. Enfin, on constate que la qualité de l'ajustement des fonctions de transfert se détériore proportionnellement à l'augmentation des paramètres NERP et DDP, c'est-à-dire lorsque l'inertie des zones augmente.

Le modèle MASH est dans l'ensemble plutôt performant pour identifier correctement les fonctions de transfert de modèles à deux zones. Deux facteurs affectent la qualité de l'ajustement : la proximité entre les jeux de paramètres et le comportement inertiel des zones. L'ajustement est d'autant meilleur que les paramètres sont différenciés et que les sous-bassins ont un comportement à faible inertie.

III.4.2 Modèles à trois branches

Le même essai est réalisé pour des modèles à trois zones. Les fonctions de transfert sont représentées **Figure III-13** et les débits **Figure III-14**. La qualité de l'ajustement de ces FT est reportée dans le **Tableau III-7**.



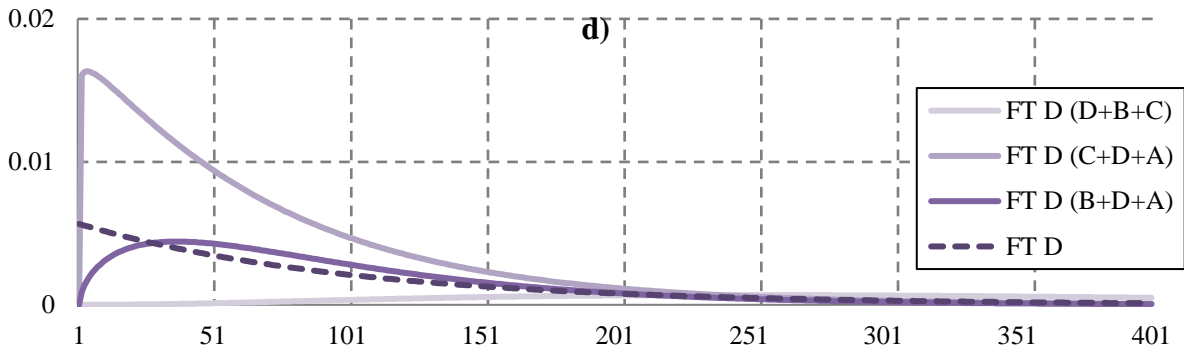
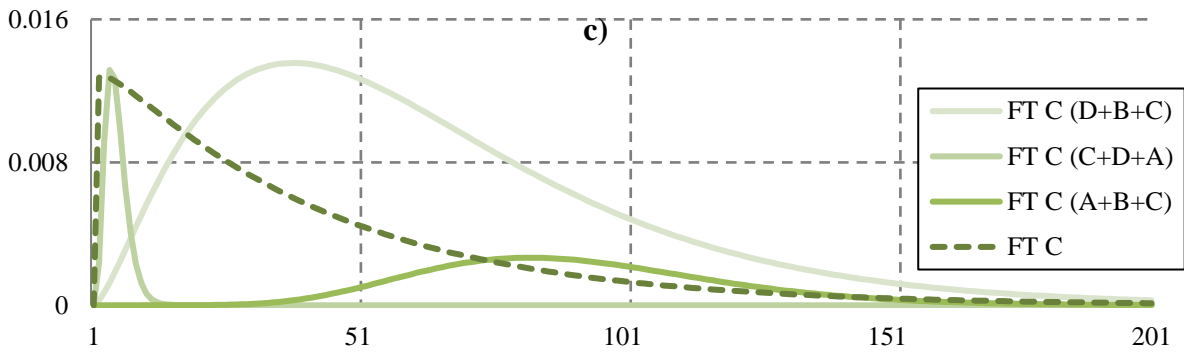
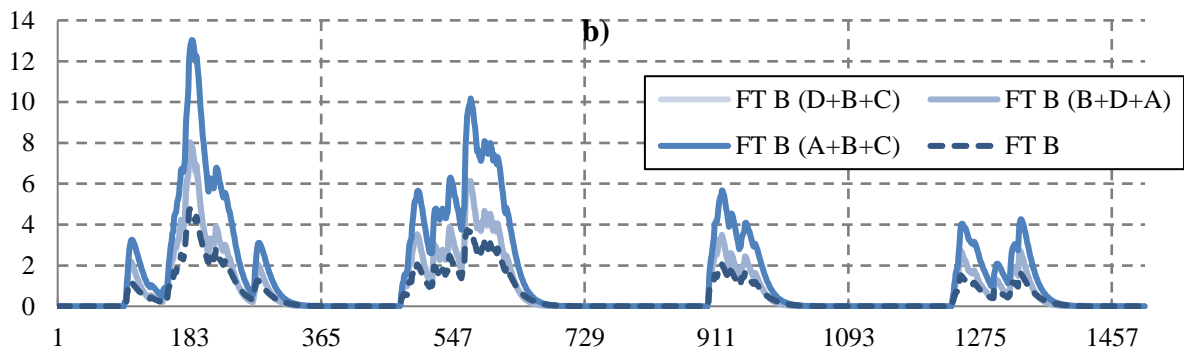
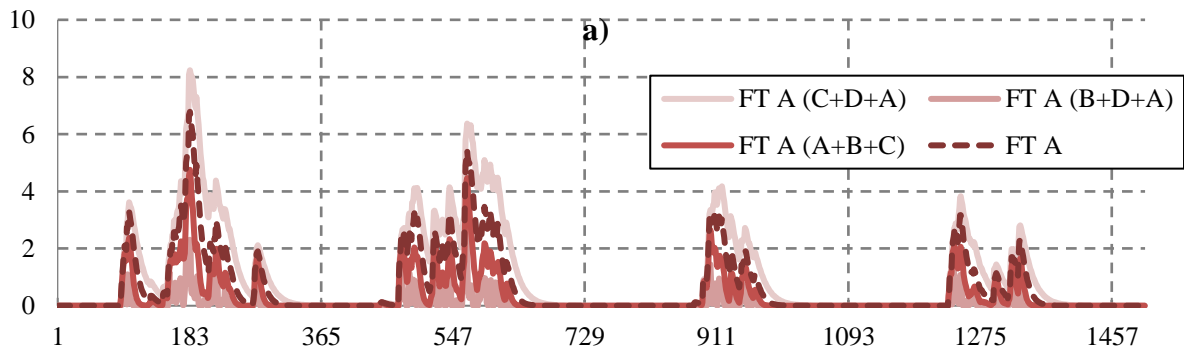


Figure III-13 : Comparaison entre les fonctions de transfert calculées pour un modèle à une zone (en trait pointillé) et celles pour un modèle à trois zones (en trait plein)



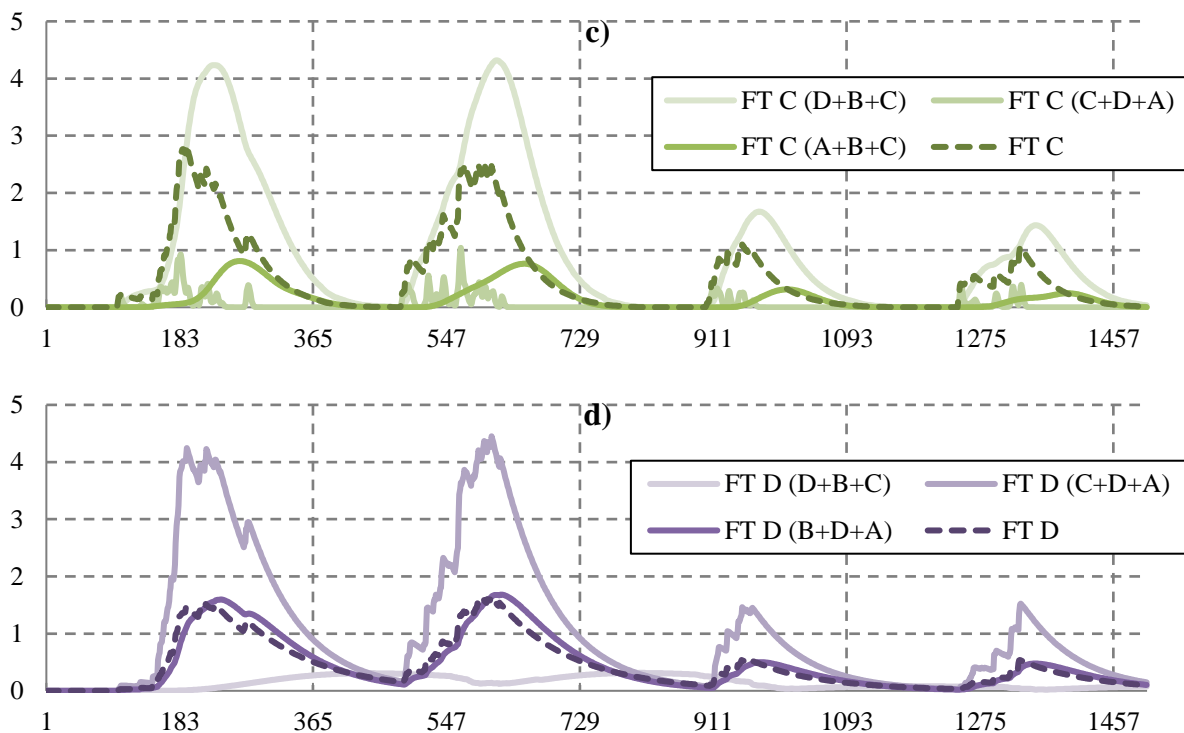


Figure III-14 : Comparaison entre les débits calculés pour un modèle à une zone (en trait pointillé) et ceux pour un modèle à trois zones (en trait plein)

Tableau III-7 : Mesure de la qualité de l'ajustement entre les fonctions de transfert des modèles MASH à une branche et ceux à trois branches

Fonction de transfert	NS
FT A (C+D+A)	0.836
FT A (B+D+A)	0.058
FT A (A+B+C)	0.805
FT B (D+B+C)	0.392
FT B (B+D+A)	0.222
FT B (A+B+C)	-3.725
FT C (D+B+C)	-1.485
FT C (C+D+A)	-0.766
FT C (A+B+C)	-1.080
FT D (D+B+C)	-1.470
FT D (C+D+A)	-7.297
FT D (B+D+A)	0.641

Dans cet exemple, seules trois fonctions de transfert donnent des résultats satisfaisants : deux sont associés à la contribution de la zone A ($NS = 0.836$ et $NS = 0.805$) et une à la contribution

de la zone D ($NS = 0.641$). Deux fonctions de transfert associées à la contribution de la zone B sont passables ($NS = 0.392$ et $NS = 0.222$), et toutes les autres sont mauvaises.

Le fait que les fonctions de transfert associées aux contributions des zones soient si éloignées des fonctions de transfert théoriques ne signifie pas que la simulation est mauvaise : pour rappel, l'ajustement des modèles à trois zones est supérieur à 0.97 pour les quatre cas testés. Cela signifie par contre que l'algorithme PSO est capable de trouver des fonctions de transfert qui offrent un bon ajustement entre le débit modélisé et le débit théorique sans que celles-ci reflètent les caractéristiques des différentes zones du bassin versant. Ce dernier point n'est guère surprenant étant donné que les problèmes que MASH essaie de résoudre sont « mal-posés », et qu'il existe un grand nombre de solutions permettant de minimiser la fonction objectif.

Les fonctions de transfert qui ne correspondent pas aux fonctions de transfert théoriques ont ainsi tendance à se compenser entre elles afin que la sortie finale soit quand même proche de la sortie théorique. Il reste néanmoins quelques points intéressants à noter. Premièrement, le fait qu'une zone soit deux fois plus grande que les autres ne la rend pas plus facile à identifier : la fonction de transfert censée correspondre à la zone B est mieux reproduite par le modèle D+B+C que par le modèle B+D+A (pour rappel, dans les modèles à trois zones, la première lettre correspond à la zone nord qui est deux fois plus grande que les autres). Deuxièmement, on constate encore une fois que l'ajustement est de meilleure qualité pour les zones bien différenciées et à faible inertie.

III.4.3 Modèles à quatre branches

Sur les **Figure III-15** et **Figure III-16** sont tracés respectivement les fonctions de transfert et les débits du modèle à quatre zones. La qualité de l'ajustement des premières est donnée **Tableau III-8**.

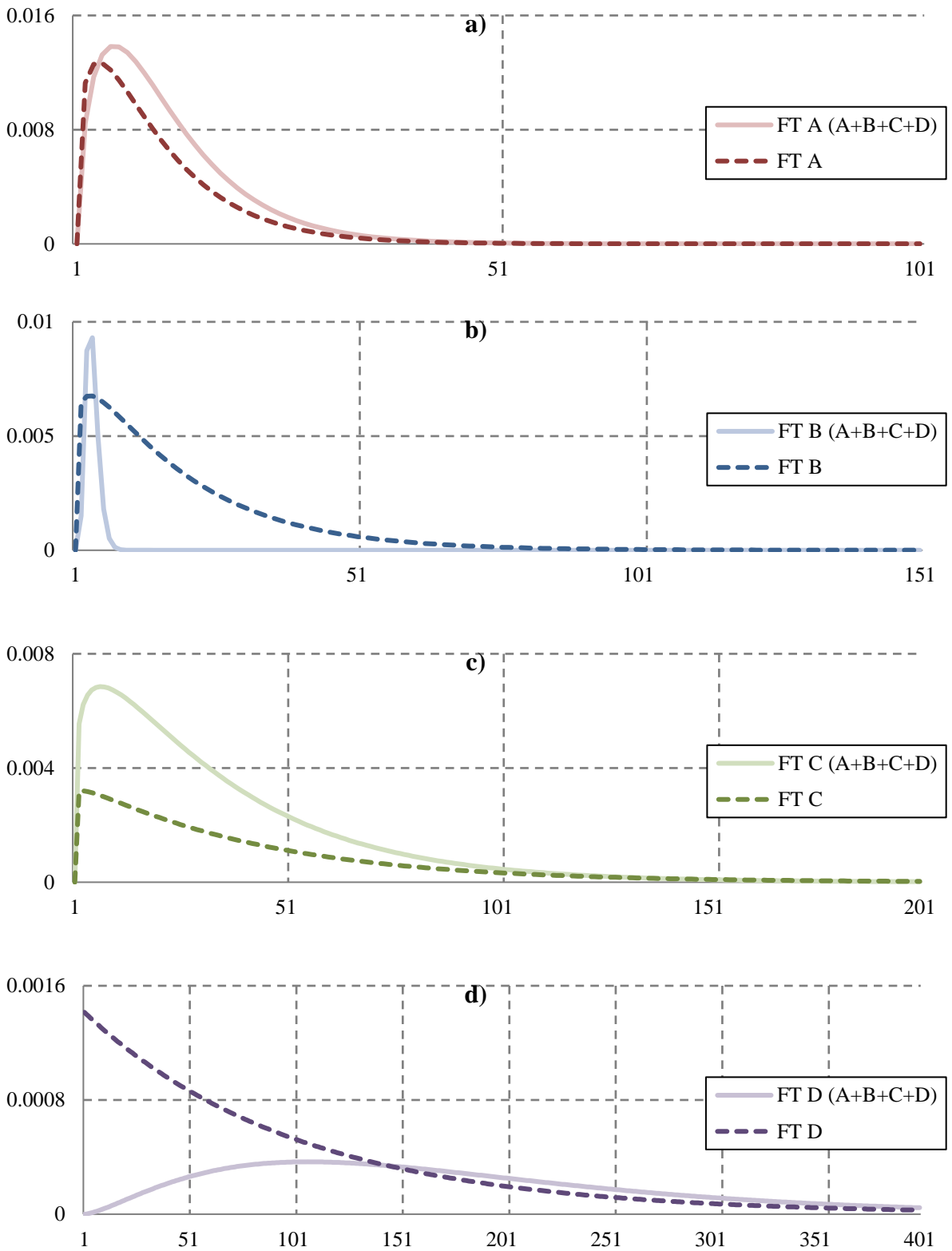


Figure III-15 : Comparaison entre les fonctions de transfert calculées pour un modèle à une zone (en trait pointillé) et ceux pour un modèle à quatre zones (en trait plein)

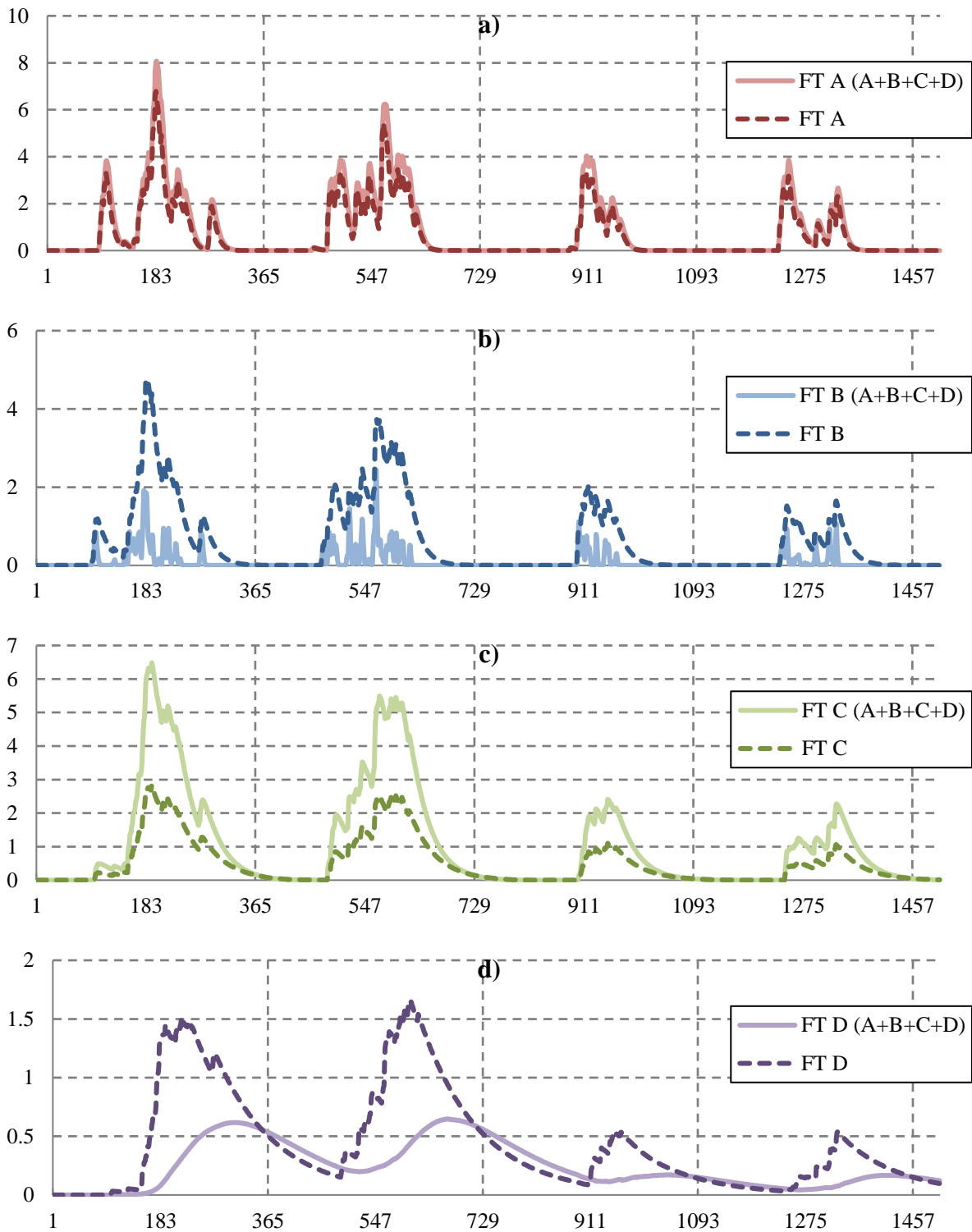


Figure III-16 : Comparaison entre les débits calculés pour un modèle à une zone (en trait pointillé) et ceux pour un modèle à quatre zones (en trait plein)

Tableau III-8 : Mesure de la qualité de l'ajustement entre les fonctions de transfert des modèles MASH à une branche et celui à quatre branches

Fonction de transfert	NS
FT A (A+B+C+D)	0.898
FT B (A+B+C+D)	-0.129
FT C (A+B+C+D)	-2.554
FT D (A+B+C+D)	-0.640

Comme pour l'exemple du modèle à trois branches, le débit total est bien restitué ($NS = 0.984$ cf. **Tableau III-5**) alors qu'individuellement les composantes censées correspondre aux différentes zones sont très différentes, exceptée pour la composante provenant de la FT A. La FT B, qui sous-estime le débit, est compensée par la FT C, qui le surestime. La FT D quant à elle n'est conforme à la fonction théorique que sur les temps longs, ce qui permet toutefois de bien simuler les périodes de longue récession.

III.5 Conclusions pour cet exemple

Les trois points importants à retenir de cette étude sont les suivants :

- MASH est capable de reproduire très efficacement les débits générés par MARTHE, pour des modèles d'une à quatre zones de sol, quels que soit les paramètres de ces différentes zones.
- Quand le modèle MARTHE est constitué de deux zones, les deux fonctions de transfert de MASH représentent de manière univoque ces différentes zones (sauf lorsque celles-ci sont trop peu différenciées ET qu'elles représentent des systèmes à forte inertie).
- Quand le modèle MARTHE est constitué de trois ou quatre zones, les trois ou quatre fonctions de transfert de MASH ne correspondent pas forcément à ce que serait les fonctions de transfert des zones individualisées (sauf lorsque celles-ci représentent des systèmes à faible inertie), ce qui montre bien la multiplicité des solutions possibles si on ajuste uniquement les flux à l'exutoire d'un système contributif à zones multiples. MASH n'échappe pas à cette multiplicité de solutions et par conséquent, toute tentative de discussion sur des éléments spécifiques des fonctions de transfert s'avère hasardeuse. Le fait peut paraître décevant, mais il en serait de même si on essayait d'inverser un modèle distribué en zones multiples (comme MARTHE et le cas test ciblé) sur les seules mesures de flux à l'exutoire. On pourrait obtenir des paramètres de zones se compensant mutuellement (et par conséquent des compensations mutuelles des zones contributives) et une solution inverse bien différente du problème de référence.

CHAPITRE IV : SIMULATION DE DONNÉES PROVENANT D'UN CAS RÉEL : LE BASSIN VERSANT DE LA SÈVRE NIORTAISE À PONT-DE-RICOU

Ce chapitre illustre un exemple d'application testant de manière exhaustive le fonctionnement du modèle MASH sur un cas réel. Il s'agit dans un premier temps de comparer les différents algorithmes d'optimisation afin d'évaluer leurs capacités d'exploration et de convergence, en fonction du nombre d'individus utilisés dans un algorithme « *particle swarm* » ainsi que de la complexité du problème d'optimisation. Afin d'étudier le comportement moyen des « essais » au cours des itérations, il est nécessaire d'effectuer des centaines de simulations, étant donné le caractère stochastique des algorithmes d'optimisation.

L'étude porte également sur la comparaison des différentes architectures qu'autorisent le modèle, et plus précisément sur l'apport éventuel des mises en série et en parallèle des fonctions de transfert. Pour tester la robustesse du modèle, cette expérience intègre une phase de calibration et une phase de prédiction. Durant la phase de calibration, les fonctions de transfert agencées selon différentes architectures sont optimisées sur une période temporelle limitée, en recherchant les jeux de paramètres permettant de reproduire au mieux la chronique de sortie sur la même période. Lors de la phase de prédiction, ces fonctions de transfert précédemment calibrées sont réutilisées sur les autres périodes temporelles, afin de quantifier la qualité des prévisions des différentes fonctions de transfert et architectures du modèle.

IV.1 Présentation de l'hydrosystème étudié

Afin d'accroître l'intérêt de cet exemple d'application, le choix de l'hydrosystème étudié s'est porté sur un bassin au fonctionnement complexe : le bassin de la Sèvre Niortaise, composé d'aquifères faillés et multicouches, avec de nombreux échanges entre les nappes et le réseau superficiel. Incidemment, on teste donc les capacités d'homogénéisation du modèle MASH dont on espère, si elles fonctionnent sur des systèmes complexes, qu'elles seront également pertinentes sur des systèmes plus simples.

La zone étudiée dans cet exemple concerne tout le bassin de la Sèvre Niortaise à l'amont de Niort, situé majoritairement dans le département des Deux-Sèvres et en partie également dans le département de la Vienne (**Figure IV-1**). Une description détaillée de ce bassin peut notamment être trouvée dans la référence [Pauwels *et al.*, 2008]. À noter que dans la partie amont, le bassin versant de la Dive du sud (topographiquement dans le bassin du Clain (Vienne)) a été ajouté car la rivière se perd dans le karst pour rejoindre la Sèvre.

Les descriptions géologique et hydrogéologique qui suivent proviennent des synthèses du BRGM, disponibles sur le site internet du SIGES (Système d'information pour la gestion des eaux souterraines) Poitou-Charentes : <http://sigespoc.brgm.fr/spip.php?article60>.

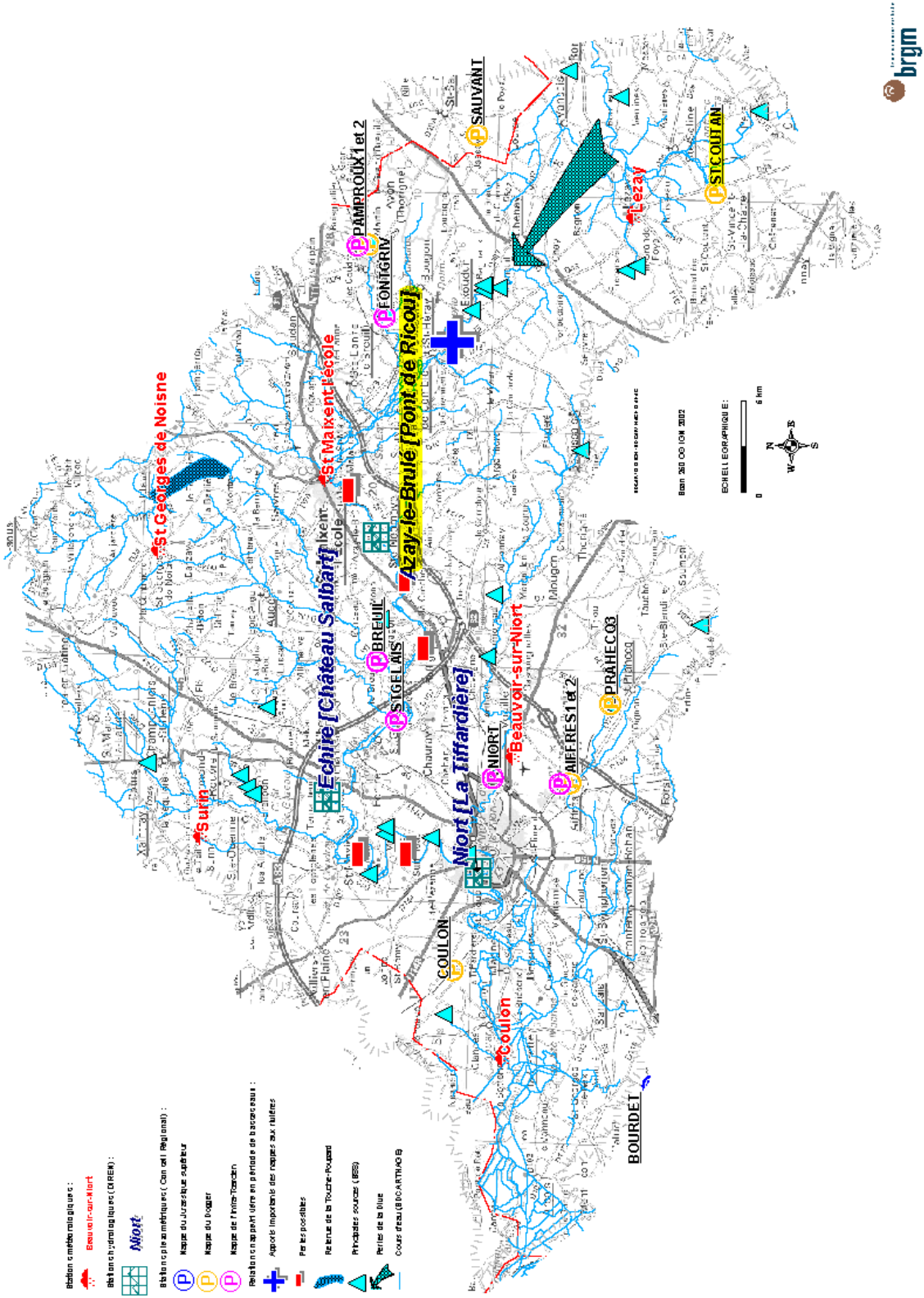


Figure IV-1 : Carte du bassin versant de la Sèvre Niortaise

IV.1.1 Géologie

D'un point de vue géologique, la vallée de la Sèvre-Niortaise est située en bordure méridionale du Massif Armoricaire. Ce socle ancien (plus de 300 millions d'années) est traversé par de nombreux accidents majeurs. La structure géologique la plus importante dans le cadre de cette étude est le système multicouche calcaire comportant les aquifères de l'Infra-Toarcien et du Dogger, séparés par un imperméable (les marnes toarciennes), et traversés par de nombreuses failles. Cette fracturation intense a favorisé le développement de la karstification.

De manière plus détaillée, la stratigraphie débute avec le socle armoricaire, formé principalement de granites, de micaschistes et de séries volcano-sédimentaires. Au-dessus du socle et de placages Infra-Lias formés d'argiles colorés et de sables, le Lias (Jurassique inférieur) est constitué sur une cinquantaine de mètres d'épaisseur de calcaires et de dolomies. Une dizaine de mètres de marnes grises du Toarcien sépare le Lias du Dogger (Jurassique moyen) sus-jacent, qui présente une sédimentation nettement carbonatée sur plus de 50 m d'épaisseur. Il s'agit principalement de calcaires en plaquettes et de calcaires oolithiques. Au-dessus, on observe un ensemble marno-calcaire du Jurassique supérieur d'environ 70 m d'épaisseur, l'ensemble du substratum jurassique étant lui-même localement recouvert par les dépôts d'altération argilo-sableuse des calcaires sur les plateaux, et par des terrasses alluviales quaternaires dans les vallées.

IV.1.2 Hydrogéologie

Plusieurs entités hydrogéologiques coexistent au sein du bassin :

Le socle est globalement imperméable mais peut renfermer dans la frange d'altération superficielle de petites nappes qui alimentent rapidement les rivières. Du fait de ce caractère peu perméable, le réseau hydrographique y est dense.

Le Lias, sous les marnes toarciennes, renferme une nappe importante. C'est un aquifère karstique assez peu épais mais pouvant développer une porosité matricielle importante. Il peut constituer des nappes libres localisées et lorsque les rivières circulent sur l'Infra-Toarcien, elles peuvent se perdre en partie dans cette nappe. Le Lias peut aussi constituer des systèmes captifs profonds compartimentés par des failles.

Au-dessus du Toarcien marneux (et de l'Aalénien), le Dogger calcaire correspond à un autre aquifère important de ce bassin. Il est souvent recouvert de formations superficielles sablo-argileuses qui peuvent emmagasiner des quantités non-négligeables d'eau progressivement diffusées vers la nappe sous-jacente. Le Dogger est aussi un aquifère particulièrement karstifié comme en témoignent les pertes de la Dive du Sud qui rejoignent la Sèvre. Ses relations avec l'aquifère infra-toarcien sont complexes mais il est évident que les deux nappes communiquent parfois entre elles, comme cela est mis en évidence par la variation des teneurs en nitrates au cours du cycle hydrologique dans des captages pour l'alimentation en eau potable où une nappe pauvre en nitrate mise en charge va alimenter l'autre nappe riche en nitrates et diluer ces derniers.

Enfin, l'Oxfordien (Jurassique supérieur) essentiellement marneux contient également une nappe superficielle à circulation essentiellement gravitaire dans une frange d'altération dont la profondeur est rarement supérieure à 30 m. Cet aquifère de milieu fissuré est peu capacitif mais sa conductivité hydraulique peut être élevée.

IV.1.3 Données

Afin de réaliser des simulations du bassin versant, le modèle MASH utilise une chronique de sortie, à savoir : la mesure du débit de la rivière, et trois chroniques d'entrée, à savoir : la pluie brute, l'ETP et un niveau piézométrique local (**Figure IV-2**). À partir des chroniques de pluie et d'ETP, MASH recalcule des chroniques de pluie nette et de pluie seuillée, selon les formules présentées paragraphes III.1.2 et III.1.3, respectivement. La période d'étude s'étend du 1^{er} aout 1993 au 31 juillet 2005, soit une durée de 4383 jours. Il s'agit de la plus grande période quasi-ininterrompue disponible pour les chroniques utilisées. Seule la chronique du piézomètre comporte quelques plages de données manquantes. Les données sont mesurées au pas de temps journalier, exceptée l'ETP fournie par Météo France et calculée au pas de temps décadaire.

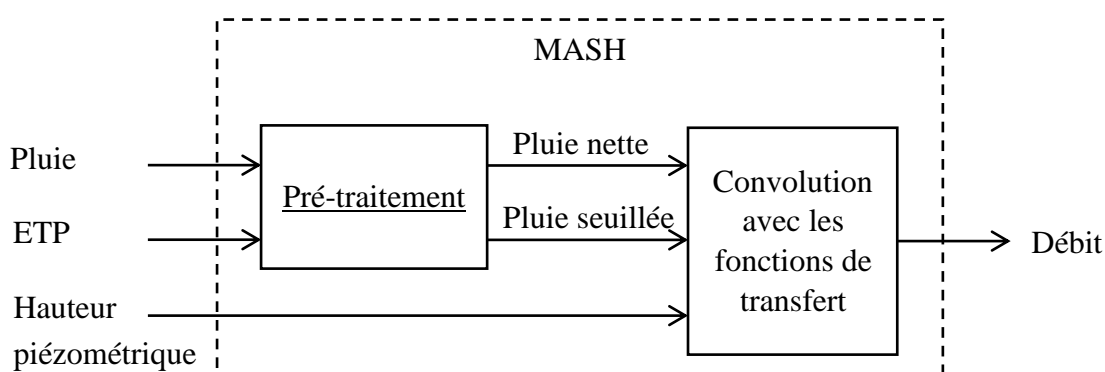


Figure IV-2 : Entrées et sorties du modèle MASH de cet exemple

Pluie

La chronique de pluie est une donnée prétraitée avec le logiciel TEMPO du BRGM. Ce programme dispose en effet d'une fonctionnalité lui permettant de calculer la chronique de pluie optimale, combinaison linéaire¹ de chroniques de pluie mesurées à plusieurs stations et qui apporte le plus d'information possible sur la chronique de sortie (ici le débit). Pour calculer les coefficients de cette combinaison linéaire, le logiciel s'efforce de maximiser l'aire du corrélogramme croisé pluie/débit [Bichot and Pinault, 2007]. Cette méthode présente des similitudes avec la méthode des polygones de Thiessen, dans le sens où elle donne un poids relatif de chacune des stations pluviométriques et représente un forçage hydrométéorologique moyenné à l'échelle du bassin versant. Les huit stations pluviométriques utilisées sont situées

¹ Plus précisément une combinaison convexe : une combinaison linéaire dont les coefficients sont positifs et de somme égale à 1.

dans les villes de Benassay, La Rochelle, La Roche-sur-Yon, Lusignan, Niort, Saintes, Rioux-Martin et Royan. Au final, l'équation utilisée est

$$P_{\text{brut}} = 0,11 \times P_{\text{Benassay}} + 0,02 \times P_{\text{La Rochelle}} + 0,23 \times P_{\text{La Roche/Yon}} + 0,14 \times P_{\text{Lusignan}} + \dots + 0,08 \times P_{\text{Niort}} + 0,24 \times P_{\text{Saintes}} + 0,08 \times P_{\text{Rioux-Martin}} + 0,1 \times P_{\text{Royan}} \quad (\text{IV.1})$$

La hauteur de pluie journalière moyenne de la chronique représentant l'ensemble des stations sur la plage de temps étudiée est de 2,44 mm, soit 890 mm par an (**Figure IV-3**).

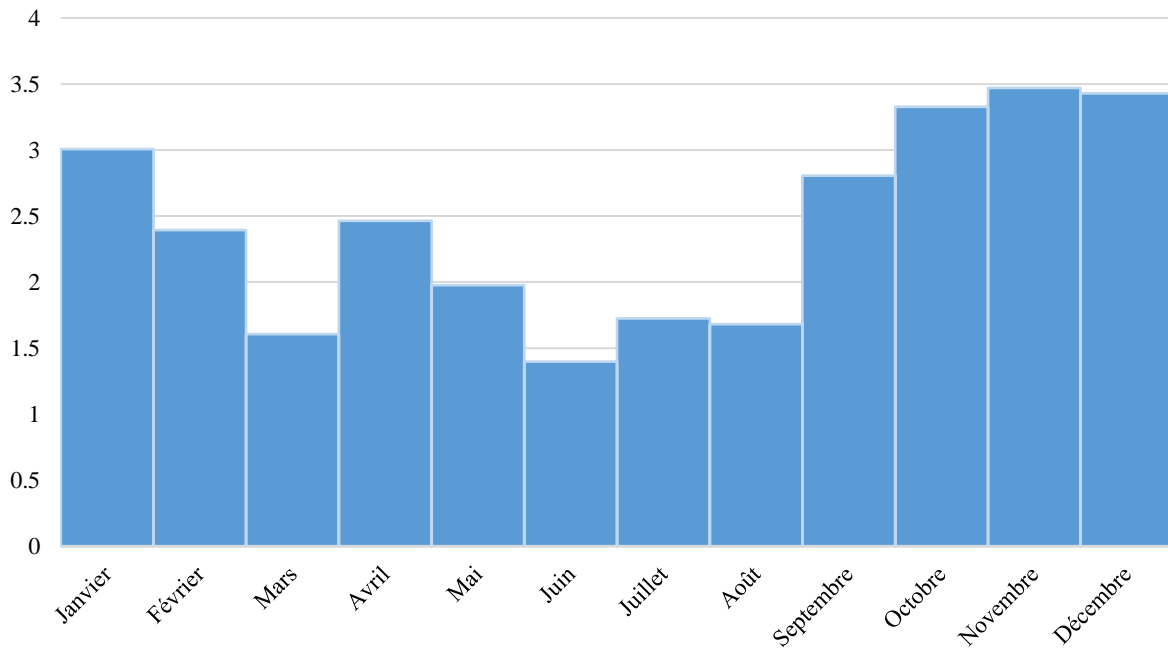


Figure IV-3 : Pluie optimale (combinaison convexe de 8 chroniques de pluie) en moyenne journalière de 1993 à 2005 (en mm)

ETP

Comme indiqué au début de ce chapitre, la chronique d'ETP (**Figure IV-4**) au pas décadaire provient de Météo France. Les paramètres ayant servi au calcul de l'ETP ont été mesurés à la station de Niort, située à proximité du centre-ville et à moins de vingt kilomètres de la station hydrométrique où est mesuré le débit. Ces paramètres de base sont : la température minimale et maximale, la vitesse moyenne du vent à 10m (ou 2m), la tension de vapeur moyenne, la durée d'insolation et le rayonnement global. L'ETP est calculée par la méthode Penman-Monteith.

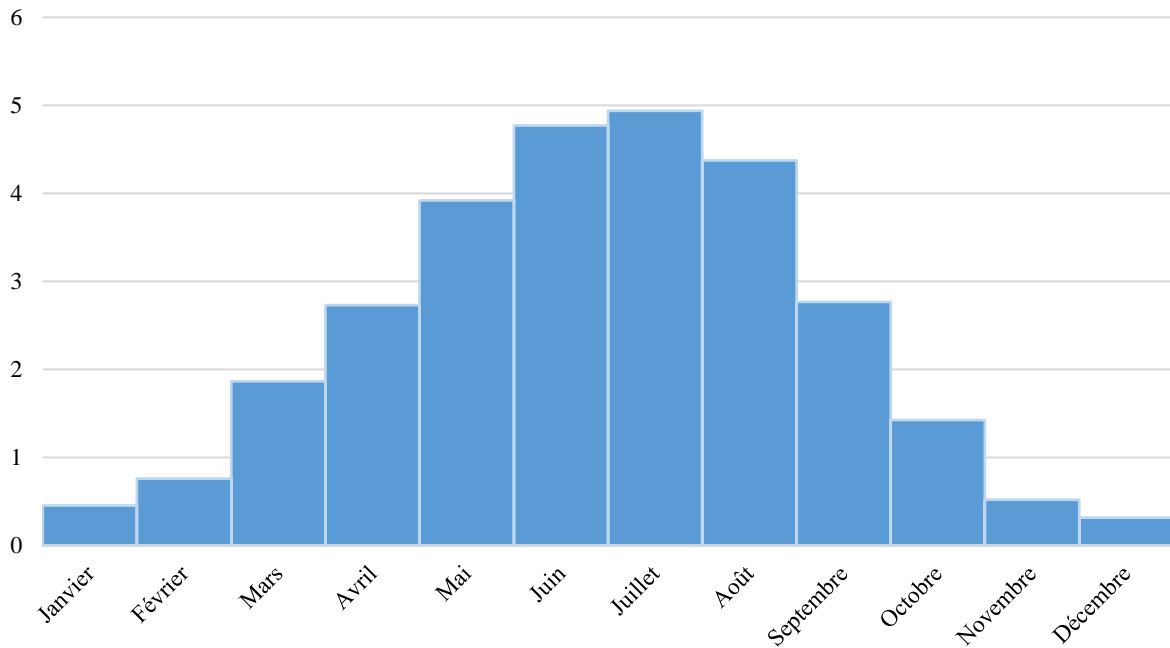


Figure IV-4 : ETP en moyenne journalière à Niort de 1993 à 2005 (en mm)

Les données sont fournies les 1^{ers}, 11^{èmes} et 21^{èmes} jours de chaque mois. Cette chronique est convertie au pas de temps journalier en amont des simulations, en utilisant la même valeur d'ETP pour 10 jours consécutifs (Figure IV-5), voire 11 jours pour les mois de 31 jours. Une interpolation linéaire aurait pu lisser la chronique, mais après des tests il s'est avéré que la qualité des simulations n'était que très peu impactée par le type d'interpolation réalisée. C'est donc la solution la moins gourmande en calcul qui a été retenue : utiliser la même valeur mesurée au jour j pour tous les jours suivants, jusqu'à la prochaine mesure au jour $j+10$.

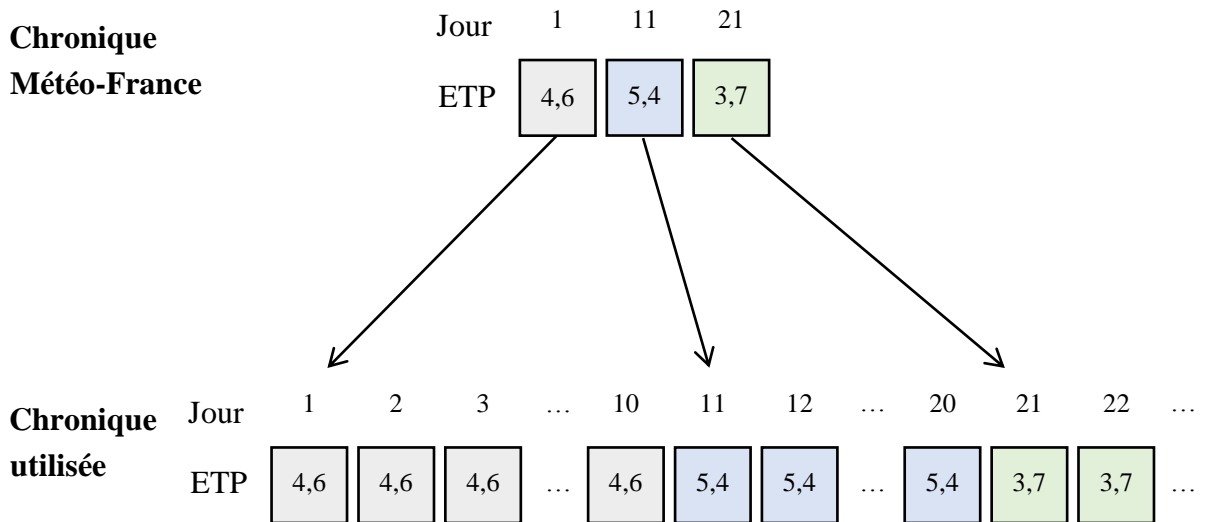


Figure IV-5 : Transformation de la chronique d'ETP

Débit

Le débit de la Sèvre Niortaise est mesuré à la station hydrométrique « Pont de Ricou » (code de la station N4010610) localisée dans la ville d'Azay-le-Brulé (79). La source de ces données est le SPC (Service de Prévision des Crues) Vienne-Charente-Atlantique. Les caractéristiques détaillées de la station sont disponibles sur le site internet de la banque Hydro (http://hydro.eaufrance.fr/selection.php?code_station=N4010610). L'année hydrologique va de septembre à août et le débit moyen mensuel est de $4,4 \text{ m}^3/\text{s}$ (Figure IV-6). Sur la période étudiée, le débit journalier maximal connu est d'environ $70 \text{ m}^3/\text{s}$.

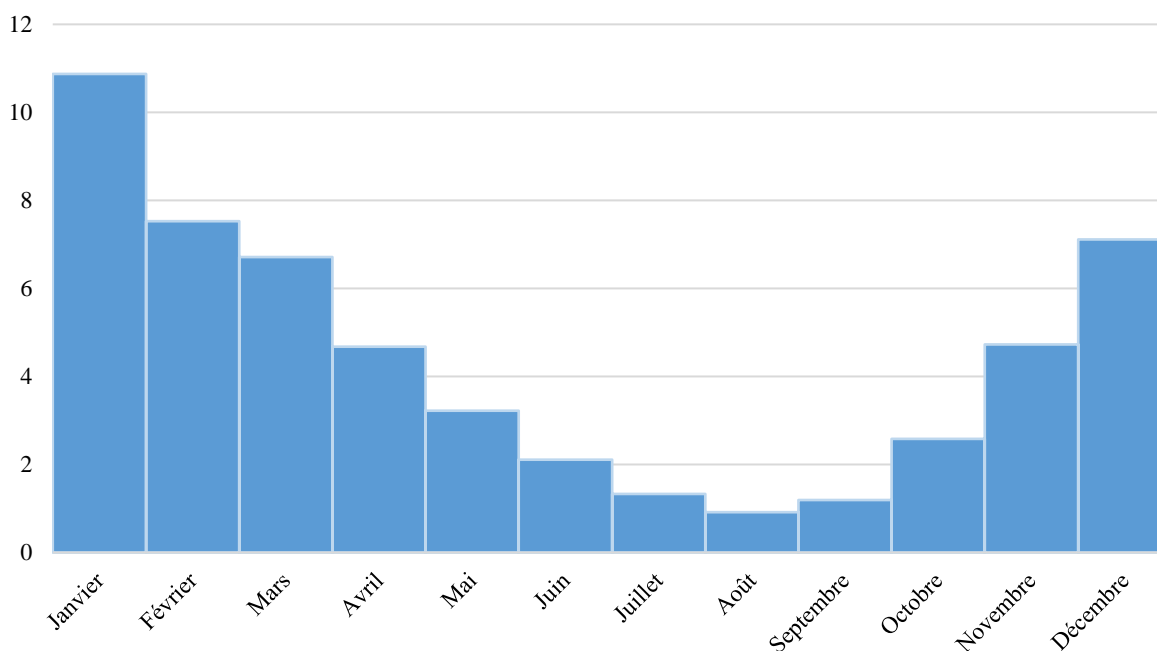


Figure IV-6 : Débit moyen mensuel de la Sèvre Niortaise à Azay-le-Brulé (Pont de Ricou) de 1993 à 2005 (en m³/s)

La fiche de synthèse de l'IRSTEA fournit également de nombreuses informations sur le bassin versant (<http://webgr.irstea.fr/wp-content/uploads/2012/06/N4010610.pdf>). Celui-ci est composé de 87% de territoires agricoles, 7% de forêt et 6% de territoires artificialisés. Le bassin versant topographique a une superficie de 240 km² (**Figure IV-7**) mais le bassin versant hydrologique est bien plus étendu, notamment à cause de la Dive-du-Sud qui disparaît après Lezay dans le karst pour rejoindre la Sèvre Niortaise à Exoudun (**Figure IV-1**).

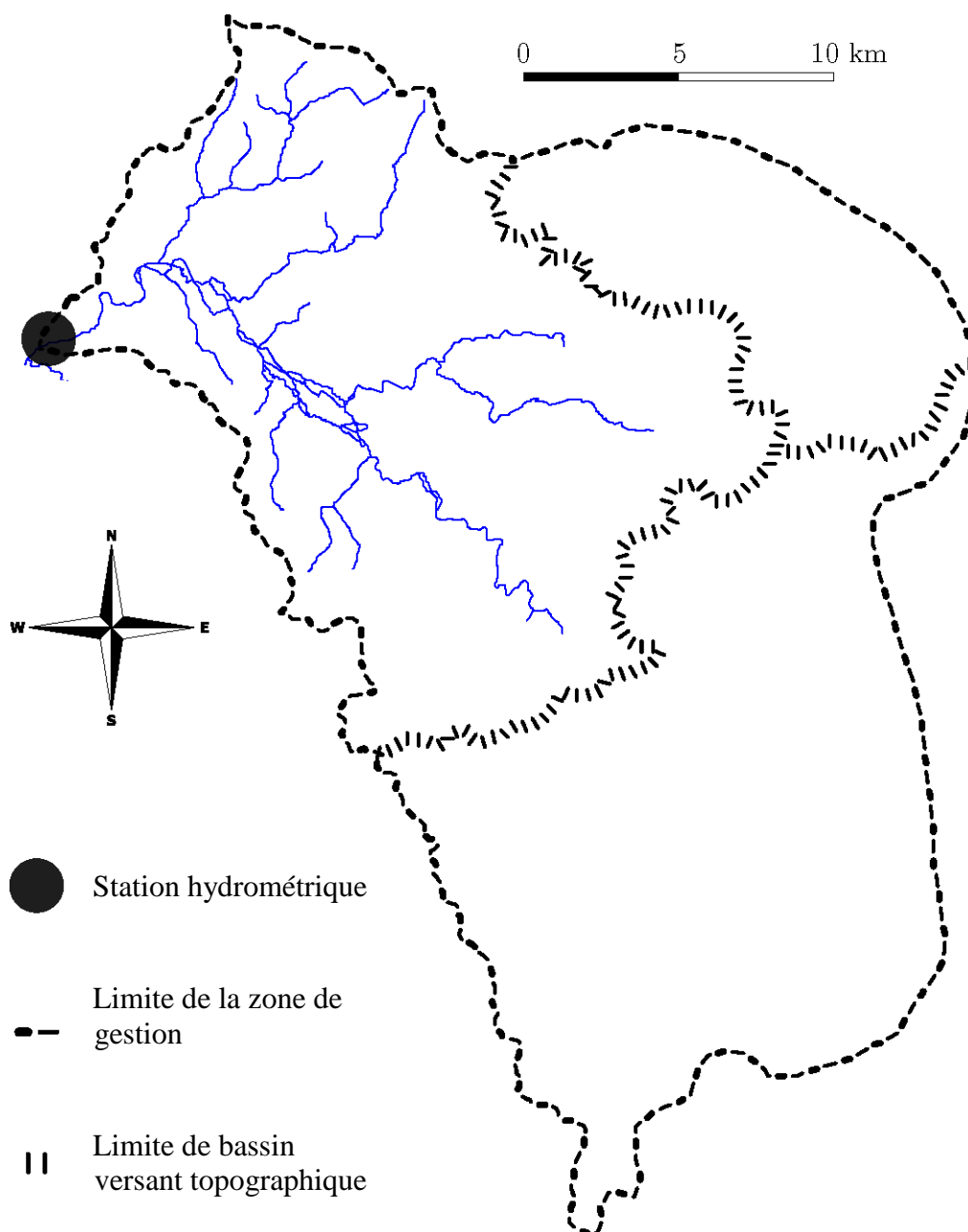


Figure IV-7 : Carte du bassin versant et localisation de la station hydrométrique Pont de Ricou. Adapté depuis [Bichot et al., 2010]

Hauteur piézométrique

La chronique de niveaux piézométriques provient du piézomètre de Saint-Coutant, (code BSS 0637-1X-0004). L'aquifère capté se situe dans les calcaires du Dogger, ayant pour code 109a2. La fiche de l'ORE (<http://www.piezo-poitou-charentes.org/P.php?n=su2&BSS=06371X0004>) présente en détail le piézomètre et sa localisation. Il a été mis en service en 1993 et est situé à environ 26 km de la station hydrométrique. Pour les simulations MASH, ce ne sont pas les valeurs piézométriques brutes qui sont utilisées mais les écarts relatifs (Figure IV-8). La valeur minimum de la série est soustraite à toutes les valeurs de la série. Ce prétraitement permet d'utiliser les mêmes fonctions de transfert que celles utilisées pour les chroniques de pluie.

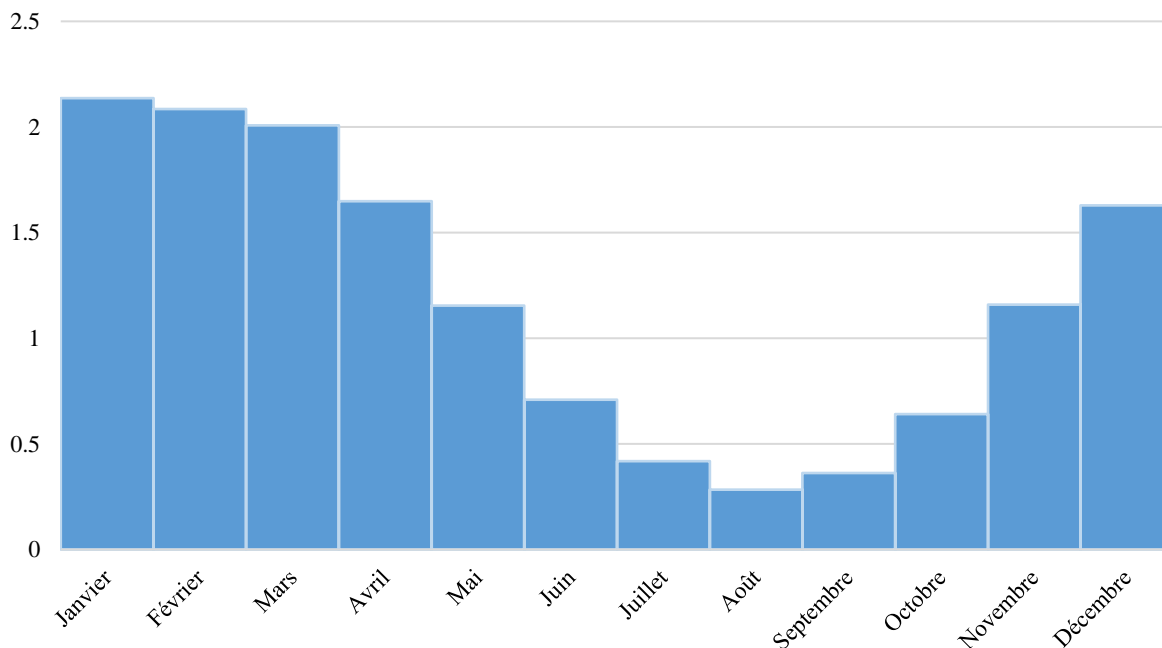


Figure IV-8 : Variation moyenne du niveau piézométrique au piézomètre de Saint-Coutant dans la nappe du Dogger de 1993 à 2005 (en m)

IV.2 Comparaison des algorithmes d'optimisation stochastiques

Une fois les données convenablement choisies, il devient possible d'étudier et de comparer le fonctionnement des trois algorithmes d'optimisation implantés dans MASH. Les modèles MASH sont inversés à l'aide de l'un des trois algorithmes d'optimisation par essaim de particules (PSO), présentés au paragraphe II.4.2 : CR-PSO (*centered-regressive PSO*), CP-PSO (*centered-progressive PSO*) et CC-PSO (*centered-centered PSO*).

Pour rappel, le principe de fonctionnement de ces algorithmes est globalement identique : à chaque itération, un essaim de particules (une particule porte le jeu de paramètres du modèle) se déplace dans l'espace des paramètres afin de trouver l'optimum de la fonction objectif. Pour cela, chaque particule garde en mémoire deux informations qui lui servent à calculer ses déplacements futurs : la meilleure position qu'elle a expérimentée, et la meilleure position expérimentée par l'essaim entier. La seule chose qui diffère entre les trois versions de l'algorithme est la manière dont les paramètres des particules sont mis à jour lors des itérations. Et c'est cette différence qui est responsable du comportement plus ou moins convergent, ou plus ou moins exploratoire de l'algorithme.

IV.2.1 Méthodologie

Trois architectures différentes de modèle ont été testées afin d'étudier le comportement des algorithmes en situation réelle. Elles sont composées respectivement de 1, 2 ou 3 fonctions de

transfert (FT) disposées en parallèle (**Figure IV-9**) pour un modèle complet à 3, 6 et 9 paramètres. Les trois entrées utilisées sont une chronique de pluie efficace, une chronique de pluie seuillée et une chronique de hauteur piézométrique. L'intégralité des chroniques est utilisée, soit 4383 valeurs par chronique. Les fonctions de transfert associées à ces entrées sont une loi Gamma pour la pluie efficace et deux lois normales pour pluie seuillée et piézométrie (les explications relatives au choix des entrées et des types de fonctions de transfert ne sont pas présentées ici, étant d'un intérêt limité pour ce test portant sur les algorithmes, mais seront détaillées au paragraphe IV.3). Pour le test, les algorithmes sont analysés avec quatre valeurs de nombre de particules : 25, 50, 100 et 200. Le nombre d'itération varie de 20 à 250, selon la rapidité de convergence de l'algorithme. Néanmoins, dans tous les cas l'optimisation continue jusqu'à la stabilisation des solutions. La qualité d'ajustement des simulations est mesurée à deux moments : lors de la 10^{ème} itération et lors de la dernière itération, afin d'étudier le comportement des algorithmes au début et à la fin de l'optimisation.

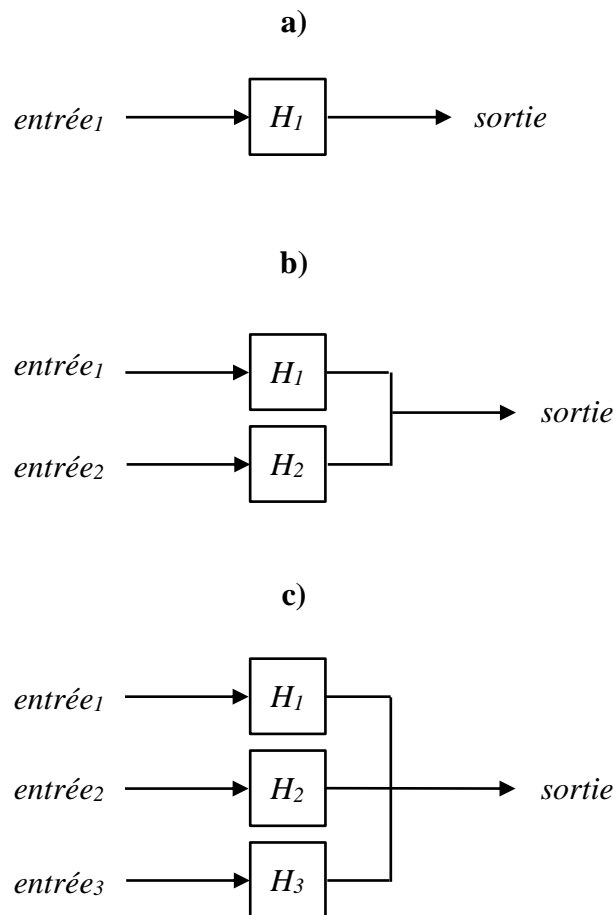


Figure IV-9 : Schéma des trois architectures MASH choisies pour comparer le comportement des algorithmes d'optimisation

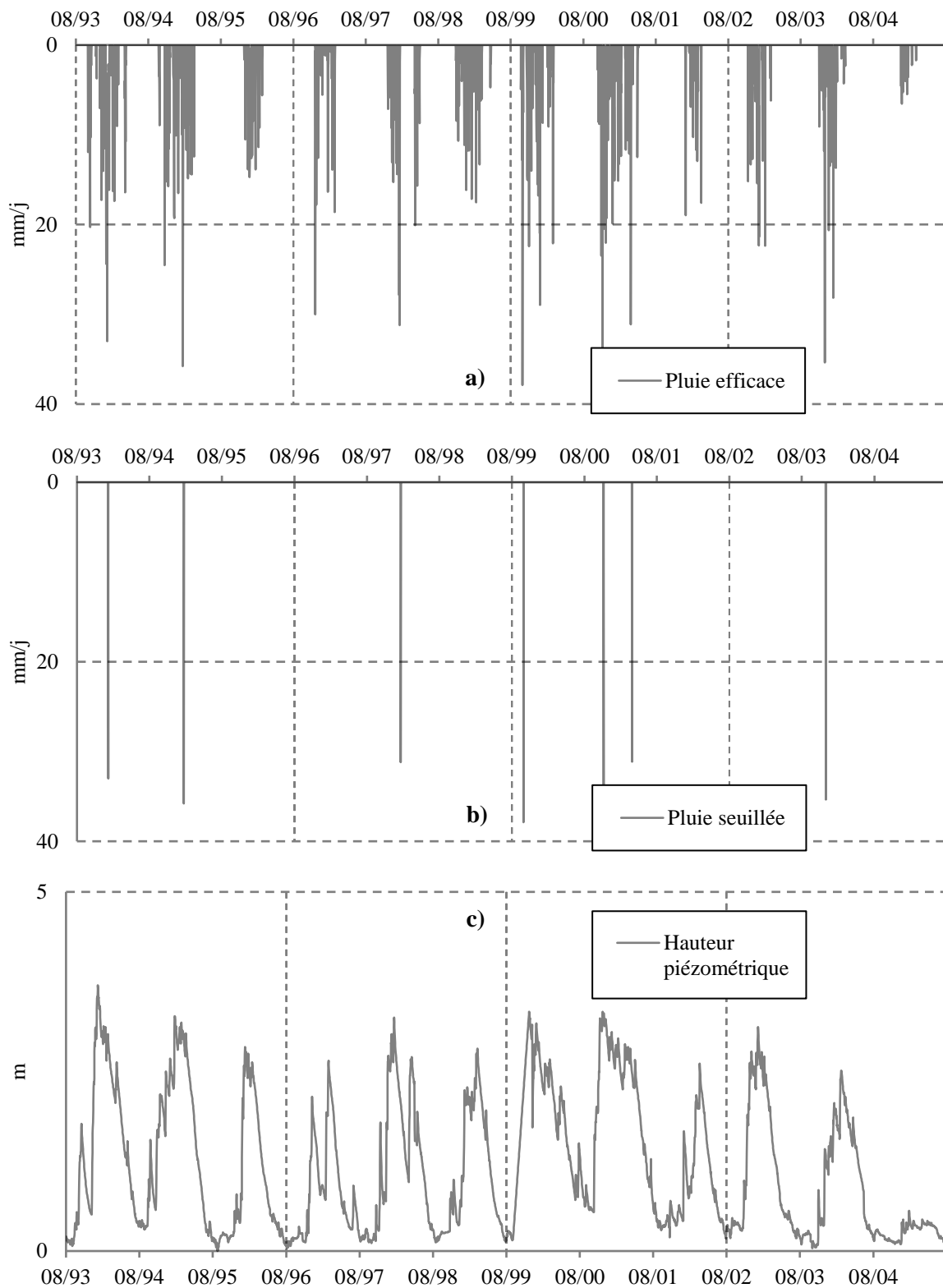


Figure IV-10 : Entrées utilisées dans les modèles (a) pluie efficace (b) pluie seuillée à 30 mm/j et (c) hauteur piézométrique.

L'étude du comportement des algorithmes ne peut se faire que de manière statistique à cause de la composante aléatoire intervenant dans l'équation de déplacement des particules. Il est

nécessaire d'effectuer un grand nombre de simulations. Ainsi pour chaque combinaison d'architecture et de nombre de particule, chacun des trois algorithmes est lancé 50 fois. À l'issue de l'optimisation, MASH fournit la valeur de la somme des écarts quadratiques entre les mesures et le modèle comme mesure de la qualité de l'ajustement. Comme chacune des optimisations est lancée 50 fois, ce sont les moyennes, les médianes et les écarts-types des ajustements (somme des écarts quadratiques) qui sont comparés.

Au total, cela représente 1800 simulations : 3 architectures, 3 algorithmes, 4 nombres de particules et 50 simulations. Les résultats sont présentés

Tableau IV-1 et **Tableau IV-2** qui proposent les mêmes résultats mais synthétisés de manière différente en les regroupant soit par architecture, soit par algorithme. Ces deux dispositions permettent d'appréhender des aspects différents des résultats de simulations.

IV.2.2 Résultats

Tableau IV-1 : Moyennes, médianes et écarts-types des ajustements (somme des écarts quadratiques) à la dixième et à la dernière itération en fonction de l'architecture du modèle.

CR : centered-regressive PSO — CP : centered-progressive PSO — CC : centered-centered PSO — Les nuances de couleur vont du vert (meilleure optimisation) au rouge (moins bonne optimisation)

			dixième itération				dernière itération				
			25 part.	50 part.	100 part.	200 part.	25 part.	50 part.	100 part.	200 part.	
1FT 3 para.	moyenne	CP	48139	44867	42246	39926	47934	44745	42080	39804	
		CC	42607	41789	39893	39022	37884	36909	36629	37263	
		CR	39670	38448	37263	36902	36558	36558	36558	36558	
	médiane	CP	47190	43967	41423	40022	47190	43967	41213	39701	
		CC	41367	39662	37736	37428	36623	36558	36558	36558	
		CR	39572	38129	37104	36847	36558	36558	36558	36558	
	écart-type	CP	5965	4224	3820	1645	6089	4328	3865	1599	
		CC	4921	6024	5002	2825	3422	1377	496	1682	
		CR	1644	1356	600	280	0	0	0	0	
	2FT 6 para.	moyenne	CP	65337	53680	45811	41336	62072	53164	45684	41327
			CC	114631	36807	27148	25223	28646	26529	21554	22451
			CR	36586	29101	24602	22998	24047	21669	20355	20266
médiane		CP	62526	55575	46410	42370	58888	55178	46410	42370	
		CC	26935	25625	21343	21188	21456	20853	20716	20578	
		CR	36459	26282	23573	21755	20336	20261	20263	20256	

	écart-type	CP	16654	8429	8259	8436	14627	7902	8076	8419
		CC	403777	26456	9739	6812	12768	12889	3721	5120
		CR	10132	8749	4204	4330	8555	5344	289	37
3FT 9 para.	moyenne	CP	111818	73635	63421	53986	84510	69196	59386	52450
		CC	184528	120242	90659	38690	30292	37797	33116	31850
		CR	43921	38144	33412	29332	23713	21508	19745	18745
	médiane	CP	94957	67470	60875	55091	74633	60938	56639	53797
		CC	49429	48317	29114	23348	25522	24073	20643	20243
		CR	42277	38314	30793	27516	20255	18953	18045	17981
	écart-type	CP	109346	24587	17652	9660	28056	20982	13506	8362
		CC	446243	187995	224413	45614	10762	51621	47926	45526
		CR	12458	10824	8264	7658	7066	6072	4464	2497

Tableau IV-2 : Moyennes, médianes et écarts-types des ajustements (somme des écarts quadratiques) à la dixième et à la dernière itération en fonction du type d’algorithme.

CR : centered-regressive PSO — CP : centered-progressive PSO — CC : centered-centered PSO — Les nuances de couleur vont du vert (meilleure optimisation) au rouge (moins bonne optimisation)

			dixième itération				dernière itération				
			25 part.	50 part.	100 part.	200 part.	25 part.	50 part.	100 part.	200 part.	
CP	moyenne	1FT	48139	44867	42246	39926	47934	44745	42080	39804	
		2FT	65337	53680	45811	41336	62072	53164	45684	41327	
		3FT	111818	73635	63421	53986	84510	69196	59386	52450	
	médiane	1FT	47190	43967	41423	40022	47190	43967	41213	39701	
		2FT	62526	55575	46410	42370	58888	55178	46410	42370	
		3FT	94957	67470	60875	55091	74633	60938	56639	53797	
	écart-type	1FT	5965	4224	3820	1645	6089	4328	3865	1599	
		2FT	16654	8429	8259	8436	14627	7902	8076	8419	
		3FT	109346	24587	17652	9660	28056	20982	13506	8362	
	CC	moyenne	1FT	42607	41789	39893	39022	37884	36909	36629	37263
			2FT	114631	36807	27148	25223	28646	26529	21554	22451
			3FT	184528	120242	90659	38690	30292	37797	33116	31850
médiane		1FT	41367	39662	37736	37428	36623	36558	36558	36558	
		2FT	26935	25625	21343	21188	21456	20853	20716	20578	

		3FT	49429	48317	29114	23348	25522	24073	20643	20243
	écart-type	1FT	4921	6024	5002	2825	3422	1377	496	1682
		2FT	403777	26456	9739	6812	12768	12889	3721	5120
		3FT	446243	187995	224413	45614	10762	51621	47926	45526
CR	moyenne	1FT	39670	38448	37263	36902	36558	36558	36558	36558
		2FT	36586	29101	24602	22998	24047	21669	20355	20266
		3FT	43921	38144	33412	29332	23713	21508	19745	18745
	médiane	1FT	39572	38129	37104	36847	36558	36558	36558	36558
		2FT	36459	26282	23573	21755	20336	20261	20263	20256
		3FT	42277	38314	30793	27516	20255	18953	18045	17981
	écart-type	1FT	1644	1356	600	280	0	0	0	0
		2FT	10132	8749	4204	4330	8555	5344	289	37
		3FT	12458	10824	8264	7658	7066	6072	4464	2497

IV.2.3 Discussion des résultats

Ces résultats permettent de décrire le comportement des algorithmes, en fonction de l'architecture, du nombre de particule, au début et à la fin des simulations.

La qualité de l'ajustement (moyennes, médianes et écart-types) est, comme on pouvait s'y attendre, proportionnel au nombre de particules utilisé. À noter que pour le modèle le plus simple (à une fonction de transfert), l'augmentation du nombre de particule n'améliore pas la qualité de l'ajustement pour l'algorithme CR, ce qui laisse à penser que celui-ci arrive à trouver systématiquement l'optimum de la fonction objectif à partir de 25 particules. Le fait d'ajouter des particules supplémentaires n'améliore plus l'optimisation.

Le nombre de fonctions de transfert (et donc de paramètres à optimiser) a aussi une influence sur la qualité de l'ajustement. Cependant cette influence n'est pas aussi simple à généraliser que celle du nombre de particules. En effet, ajouter des paramètres à optimiser complexifie le modèle, l'espace des solutions gagnant une dimension à chaque paramètre ajouté, ce qui peut engendrer un moins bon ajustement si l'algorithme d'optimisation peine à trouver une solution dans cette espace plus vaste. Cependant, cet effet est limité par le fait qu'en ajoutant une nouvelle entrée et une nouvelle fonction de transfert, de nouvelles informations² et de nouveaux degrés de liberté apparaissent, permettant aux algorithmes de trouver de meilleures solutions. Ici, le fait d'ajouter des fonctions de transfert dégrade les convergences lors des premières itérations, ce qui reste logique puisque le problème a plus de degrés de liberté. Pour la dernière itération, la situation est plus contrastée : l'algorithme CP donne de meilleurs résultats pour le

² La troisième entrée apporte moins d'informations que la seconde : l'amélioration des ajustements entre les meilleurs modèles (CR avec 200 particules) est de 80% entre les architectures à une et deux entrées, alors qu'elle est de seulement 10% entre les architectures à deux et trois entrées.

modèle à une FT, l'algorithme CC pour le modèle à deux FT et l'algorithme CR pour le modèle à trois FT, sauf au niveau des écarts-types qui sont toujours les plus faibles pour les modèles à une FT (moins de degrés de liberté font converger systématiquement vers la même solution, éventuellement mauvaise en termes d'ajustement des données).

L'algorithme CR donne systématiquement les meilleures valeurs à la dernière itération sur les indicateurs moyennes, médianes et écarts-types. Ses performances au début de l'optimisation sont aussi excellentes à la dixième itération où CR est seulement surpassé par l'algorithme CC pour les médianes des ajustements des modèles à 2 et 3 FT.

L'algorithme qui se place en seconde position est le CC. Il optimise globalement mieux les paramètres que le CP, sauf pour les écarts-types du modèle à trois FT où l'algorithme CP donne des solutions plus proches les unes des autres (mais dont les médianes et moyennes sont moins bonnes).

Afin d'expliquer les différences de comportement de ces différents algorithmes, il est possible d'étudier plus en détail l'évolution de l'ajustement en fonction des itérations. En annexe sont tracées des courbes indiquant la médiane, les 1^{er} et 3^{ème} quartiles ainsi que le minimum et le maximum des ajustements pour les 50 exécutions, en fonction des itérations, pour chaque algorithme, chaque architecture et chaque nombre de particules (Figure B-1 à Figure B-12). Ces courbes permettent d'observer la vitesse de convergence des algorithmes : le CP-PSO converge très rapidement vers une solution souvent suboptimale ; le CC-PSO et le CR-PSO convergent plus lentement, mais vers de meilleures solutions.

Pour représenter graphiquement le caractère exploratoire ou convergent des essais, il a fallu comparer les paramètres des fonctions de transfert obtenus par les différentes versions de l'algorithme d'optimisation (**Figure IV-11**) à la valeur optimale de la fonction objectif existant dans l'espace des paramètres (**Figure IV-12**). Un modèle à trois paramètres (i.e. une FT) a été exécuté 500 fois pour chacun des algorithmes, avec 20 particules, toujours avec les mêmes données. La position dans l'espace des solutions de toutes ces particules, pour les 500 exécutions, est projetée dans un plan ayant pour abscisse le premier paramètre du modèle, et pour ordonnée le deuxième paramètre (**Figure IV-11**). Les positions des particules sont relevées à certaines itérations clefs : 1, 5 et 10 pour observer le début de l'optimisation ; 25 pour le milieu de l'optimisation ; 50 pour la fin (cf. **Figure B-1** : l'ajustement ne varie quasiment plus après 50 itérations). L'essaim de particule est alors visible et il devient possible d'observer son évolution. À l'initialisation, les paramètres sont tirés de manière aléatoire entre 1 et 10, c'est pourquoi à l'itération 1 la majorité des particules est encore située dans l'intervalle $[1,10]^2$. Étant donnée la simplicité de ce modèle (seulement trois paramètres), il est aisé d'étudier la géométrie de la fonction objectif. Pour cela, sa valeur a été calculée de très nombreuses fois en faisant varier les paramètres 1, 2 et 3 (de 0,2 en 0,2) avec les mêmes données, afin de construire une représentation fidèle de sa géométrie. Celle-ci est convexe et possède donc un minimum global. Les coordonnées de ce minimum sont (18,12 ; 1,01 ; 3,67), respectivement pour les paramètres 1, 2 et 3. C'est pour cela que les projections des **Figure IV-11** et **Figure IV-12** sont faites dans le plan $paramètre3 = 3,67$. On constate dans la **Figure IV-12** que le minimum de la fonction objectif est situé dans une vallée étroite parallèle à l'axe *paramètre1*, ce qui explique pourquoi les particules de la **Figure IV-11** convergent rapidement selon le paramètre 2 et plus lentement selon le paramètre 1. Ce phénomène aurait pu être minimisé en normalisant ces paramètres (et

à condition que le modèle soit sensible aux paramètres), mais pas complètement supprimé, à cause de la forme de la fonction de transfert.

L'algorithme CP-PSO (**Figure IV-11, a**) est le plus exploratoire. Alors que la majorité des particules de la première itération se trouve encore dans la fenêtre tracée $[-10,30] \times [-5,15]$, les particules se dispersent ensuite de plus en plus au fur et à mesure des itérations. En faisant de grands sauts, les particules arrivent à explorer une partie très vaste de l'espace des solutions, mais la majorité passe à côté de la vallée contenant le minimum global (**Figure IV-12**). L'algorithme CP est donc trop exploratoire pour ce problème.

À l'inverse, les particules du CC-PSO convergent rapidement (**Figure IV-11, b**), cet algorithme étant le plus convergent. À l'itération 1, les essaims sont toujours très contenus dans l'intervalle de départ. À l'itération 5, une partie des particules commence à se diriger vers la vallée du minimum global. Ce mouvement continue jusqu'à l'itération 25, au-delà de laquelle les particules ont atteint leur position finale. Le paramètre 2 est identifié (égal à 1.0) par la majorité des particules, alors que le paramètre 1 (égal à 18.1) n'est trouvé que par un faible pourcentage d'entre elles. L'algorithme CC-PSO converge donc trop vite pour ce problème : les essaims peinent à trouver des solutions situées au-delà de leur espace de départ car ils se resserrent rapidement dès qu'un de leur agent trouve une solution intéressante.

L'algorithme CR-PSO est le plus modéré : il est plus convergent que le CP-PSO, mais plus exploratoire que le CC-PSO (**Figure IV-11, c**). Cela permet aux essaims d'explorer l'espace des solutions sans rester bloqué dans l'espace de départ et sans se perdre dans des zones inintéressantes. Ainsi, les particules évoluent plus lentement vers le minimum global et continuent de converger pendant de nombreuses itérations, même à la fin de l'optimisation. Dès l'itération 25, le paramètre 2 est largement identifié par la plupart des particules et le paramètre 1 commence à être cerné. À l'itération 50, cette tendance est confirmée, la grande majorité des essaims sont centrés sur le point (1.0,18.1).

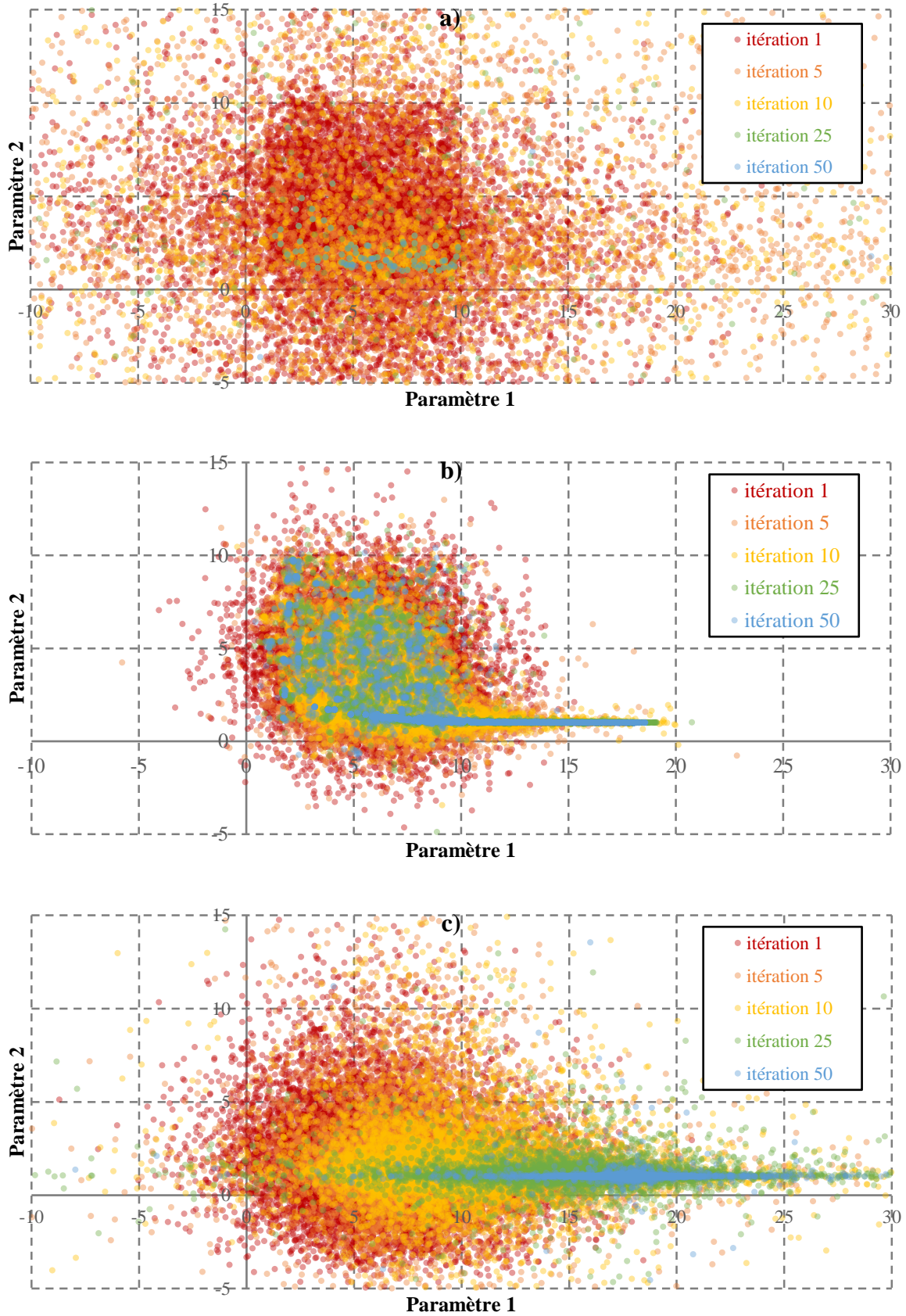


Figure IV-11 : Projection de la position des particules aux itérations 1, 5, 10, 25 et 50, pour 500 exécutions avec 20 particules des algorithmes (a) CP-PSO (*centered-progressive PSO*) ; (b) CC-PSO (*centered-centered PSO*) ; (c) CR-PSO (*centered-regressive PSO*)

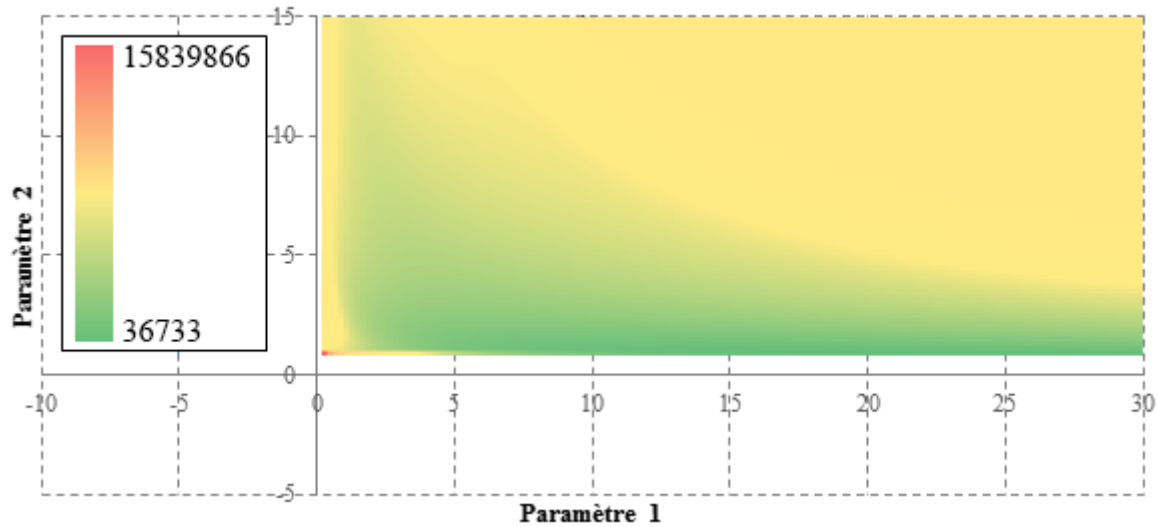


Figure IV-12 : Carte de chaleur de l'ajustement (somme des écarts quadratiques) pour le modèle à trois paramètres, en fonction des paramètres 1 et 2 (paramètre 3 = 3.67) (les zones blanches correspondent aux zones où la fonction de transfert n'est pas définie).

Pour ce problème, l'algorithme CR-PSO surpasse systématiquement les deux autres. C'est donc celui-là qui sera utilisé par la suite.

Pour choisir les paramètres de l'algorithme (nombre de particules et nombre d'itérations) en fonction du nombre de paramètres, il faut noter que vu l'architecture de l'algorithme, le temps de calcul est proportionnel au nombre d'appels du modèle direct, car c'est de loin l'opération la plus longue à effectuer. Comme ce nombre d'appels est égal au nombre de particules multiplié par le nombre d'itérations, il est nécessaire de trouver un compromis entre nombre de particules et nombre d'itérations³. Pour les modèles les plus simples, le nombre d'itérations est privilégié par rapport au nombre de particules car l'algorithme converge bien même pour un petit essaim. Pour les modèles plus complexes, c'est la taille de l'essaim qui est privilégiée afin de s'assurer que les solutions sont proches de la solution optimale. Dans tous les cas, les optimisations sont toujours exécutées au moins trois fois et c'est la meilleure exécution qui est conservée.

IV.3 Comparaison des architectures du modèle

Après avoir décrit les données et comparé les algorithmes d'optimisation, la suite de ce chapitre porte sur l'étude du fonctionnement du modèle MASH, afin de tenter de répondre aux trois questions suivantes :

³ À titre indicatif, avec l'ordinateur utilisé, équipé d'un processeur AMD Athlon 64 Dual Core 6000+ cadencé à 3.00GHz, la durée de calcul est d'environ 4.7 ms/(particule*itération). Une optimisation dure donc 11,75 s en utilisant 25 particules durant 100 itérations ou 100 particules durant 25 itérations.

1. Quelles fonctions de transfert et quelle architecture génèrent les meilleures simulations ?
2. Les modèles plus complexes donnent-ils de meilleurs résultats que les modèles simples ?
3. Ces modèles sont-ils performants, i.e. l'ajustement est-il aussi bon en calibration qu'en prédiction ?

Afin de répondre aux deux premières questions, le modèle MASH est analysé en choisissant des architectures de plus en plus complexes pour simuler un même jeu de données. Cela permet de tester si d'une part certaines fonctions de transfert donnent de meilleures simulations que d'autres, et si d'autre part les modèles complexes simulent mieux que les modèles simples (réponse courte : ce n'est pas forcément le cas). Pour répondre à la troisième question, tous les tests sont faits en validation croisée pour vérifier si les modèles sont aussi performants en calibration qu'en prédiction.

IV.3.1 Méthodologie

Le modèle MASH peut posséder jusqu'à quatre branches, chacune constituée de trois fonctions de transfert, elles-mêmes sélectionnées dans une bibliothèque de quatre fonctions. Le nombre de combinaisons d'architectures à tester est donc trop élevé pour pouvoir réaliser une étude exhaustive, d'autant que toutes les architectures sont analysées en validation croisée. Afin de disposer néanmoins d'une vue d'ensemble quant à la qualité des ajustements correspondants aux différentes architectures et fonctions de transfert possibles, les modèles testés ont des architectures de plus en plus complexes mais conservent les fonctions de transfert ayant donné les meilleurs résultats pour des modèles plus simples. Également dans un souci de simplicité, seul un type d'algorithme PSO sera testé. C'est le CR-PSO qui sera utilisé car c'est l'algorithme qui s'est avéré être le plus performant pour ce jeu de données.

Architecture à une branche

L'architecture la plus simple est celle à une branche et une seule fonction de transfert. Les quatre types de fonctions sont testées l'une après l'autre pour identifier celle qui donne le meilleur ajustement (**Figure IV-13, a**). Une fois la meilleure fonction identifiée, une deuxième fonction de transfert est ajoutée. La première FT est celle qui a été identifiée précédemment ; pour la seconde FT, les quatre types de fonction sont à nouveau testés successivement (**Figure IV-13, b**). L'architecture à trois FT en série est étudiée exactement de la même manière en reprenant les deux meilleures fonctions de transfert du modèle précédent (**Figure IV-13, c**). Pour les architectures à deux et trois FT, même si la forme de certaines fonctions est sélectionnée, tous les paramètres de toutes les FT sont néanmoins recalibrés, les fonctions en série créant des corrélations entre paramètres.

La chronique de sortie est la même pour toutes les architectures : il s'agit de la chronique de débit, introduite paragraphe 0. La chronique utilisée en entrée est une chronique de pluie nette calculée avec les chroniques de pluie et d'ETP présentées respectivement aux paragraphes 0 et 0 selon la formule de la fonction de production (paragraphe II.1.2). Pour ce modèle à une

branche, il est nécessaire de sélectionner la chronique apportant le plus d'information permettant de simuler la sortie. Le corrélogramme croisé entre la sortie et les différentes entrées (**Figure IV-14**) apporte en effet la preuve que le signal le plus corrélé avec la sortie est la chronique de pluie.

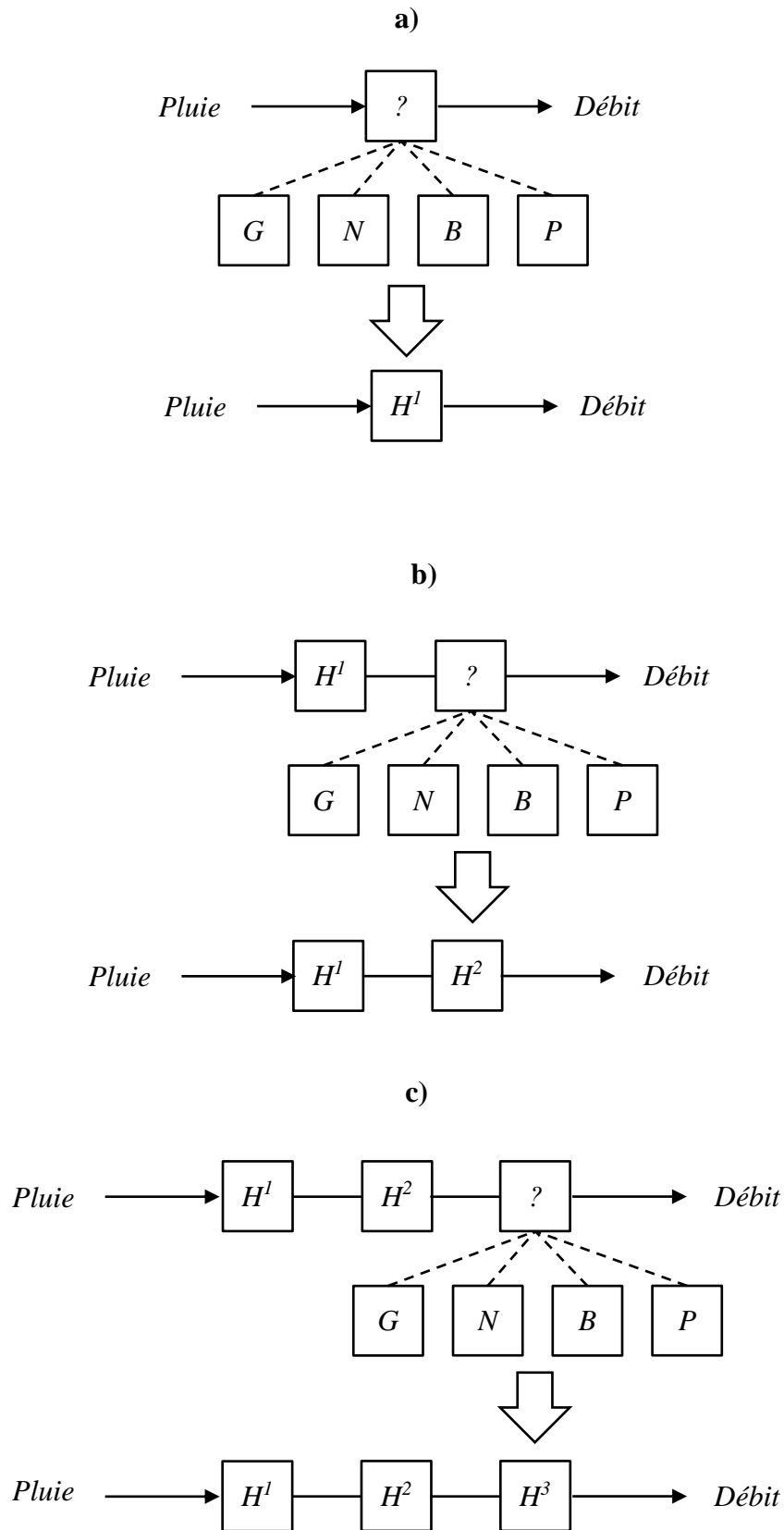


Figure IV-13 : Recherche de la meilleure fonction de transfert pour un modèle à une branche et (a) une, (b) deux et (c) trois fonctions de transferts.
 (G : loi Gamma – N : loi normale – B : loi Béta – P : loi puissance)

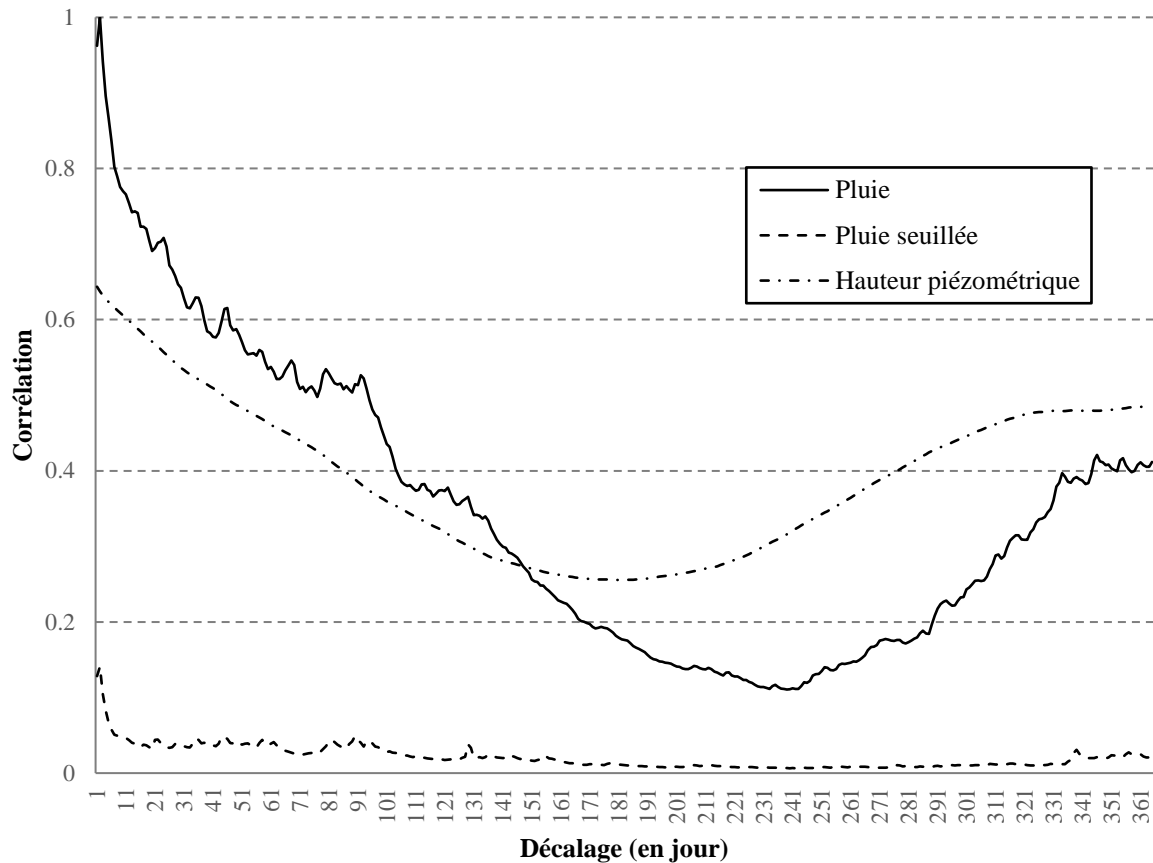


Figure IV-14 : Corrélogrammes croisés entre la sortie (Débit) et les différentes entrées (Pluie, Pluie seuillée et Hauteur piézométrique)

Architecture à deux branches

Le déroulement des tests pour l'architecture à deux branches parallèles reste similaire. L'architecture de départ comporte une fonction de transfert par branche (Figure IV-15, a). Les quatre types de fonction sont testés pour chacune des branches, soit seize possibilités. Comme précédemment, une fois la meilleure combinaison identifiée, une seconde (Figure IV-15, b) puis une troisième (Figure IV-15, c) fonction de transfert sont ajoutées à chaque branche et les seize combinaisons de FT sont de nouveau testées. L'entrée de la seconde branche est la chronique

de hauteur piézométrique. C'est le deuxième signal le plus corrélé avec la chronique de sortie (Figure IV-14).

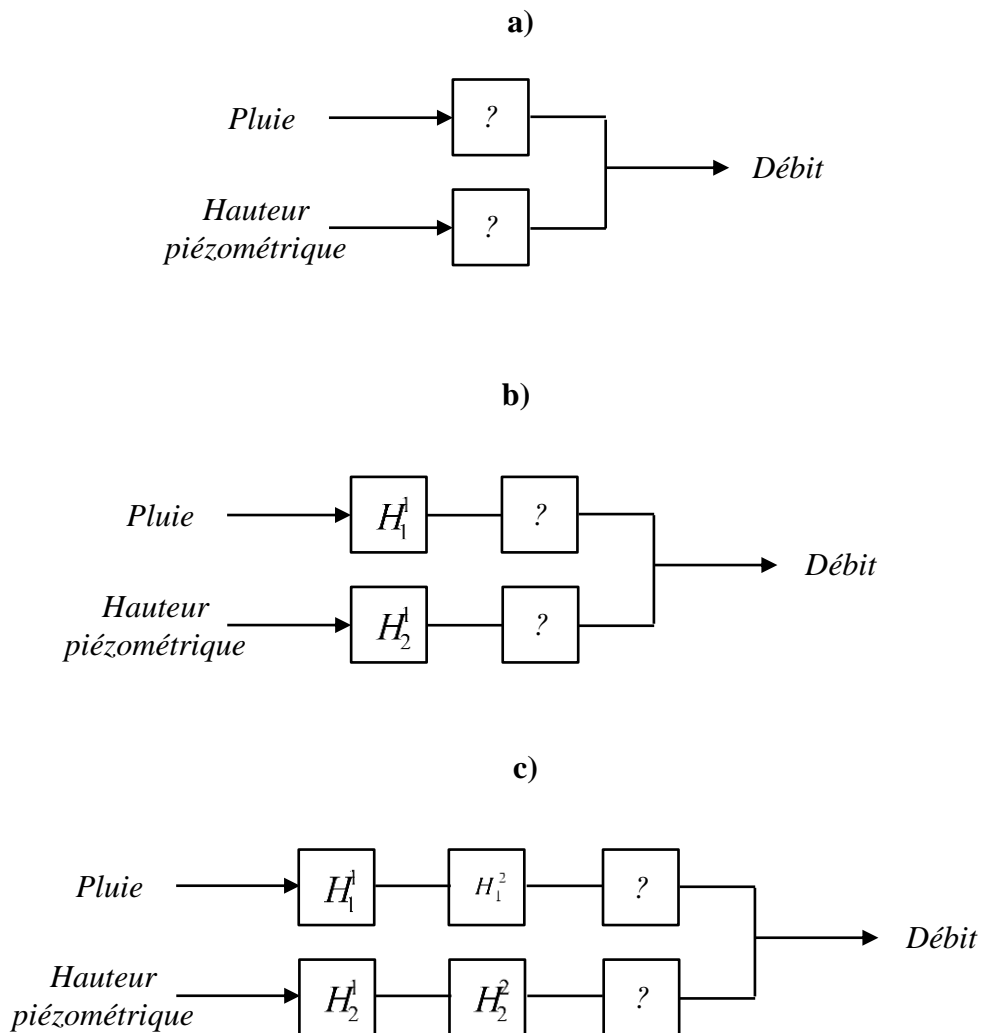


Figure IV-15 : Recherche de la meilleure fonction de transfert pour un modèle à deux branches et (a) une, (b) deux et (c) trois fonctions de transferts par branche.

Architecture à trois branches

L'architecture à trois branches est très fortement empreinte de celle à deux branches. En effet, pour la troisième branche, la fonction de transfert est toujours la même : la loi normale. Pour l'exemple analysé, même en utilisant trois fonctions de transfert en série dans cette troisième branche, la fonction de transfert équivalente a toujours une forme simple : celle d'un Dirac décalé dans le temps. Autant donc ne mettre qu'une seule fonction de transfert dans cette branche afin de limiter le nombre de paramètres à optimiser.

La simplicité de la fonction de transfert vient probablement du fait que la troisième entrée apporte relativement peu d'information (**Figure IV-14**). Il s'agit en effet d'une chronique de pluie seuillée, créée selon la méthode décrite paragraphe II.1.3 **Erreur ! Source du renvoi introuvable.** où seules les journées où la pluie nette est supérieure à 30mm sont prises en compte. Cette chronique permet d'améliorer la simulation de débits successifs à de fortes précipitations. Intuitivement, elle correspond à un comportement différent de l'hydrosystème lors des périodes de précipitation intense, comme la genèse d'un écoulement hypodermique intense.

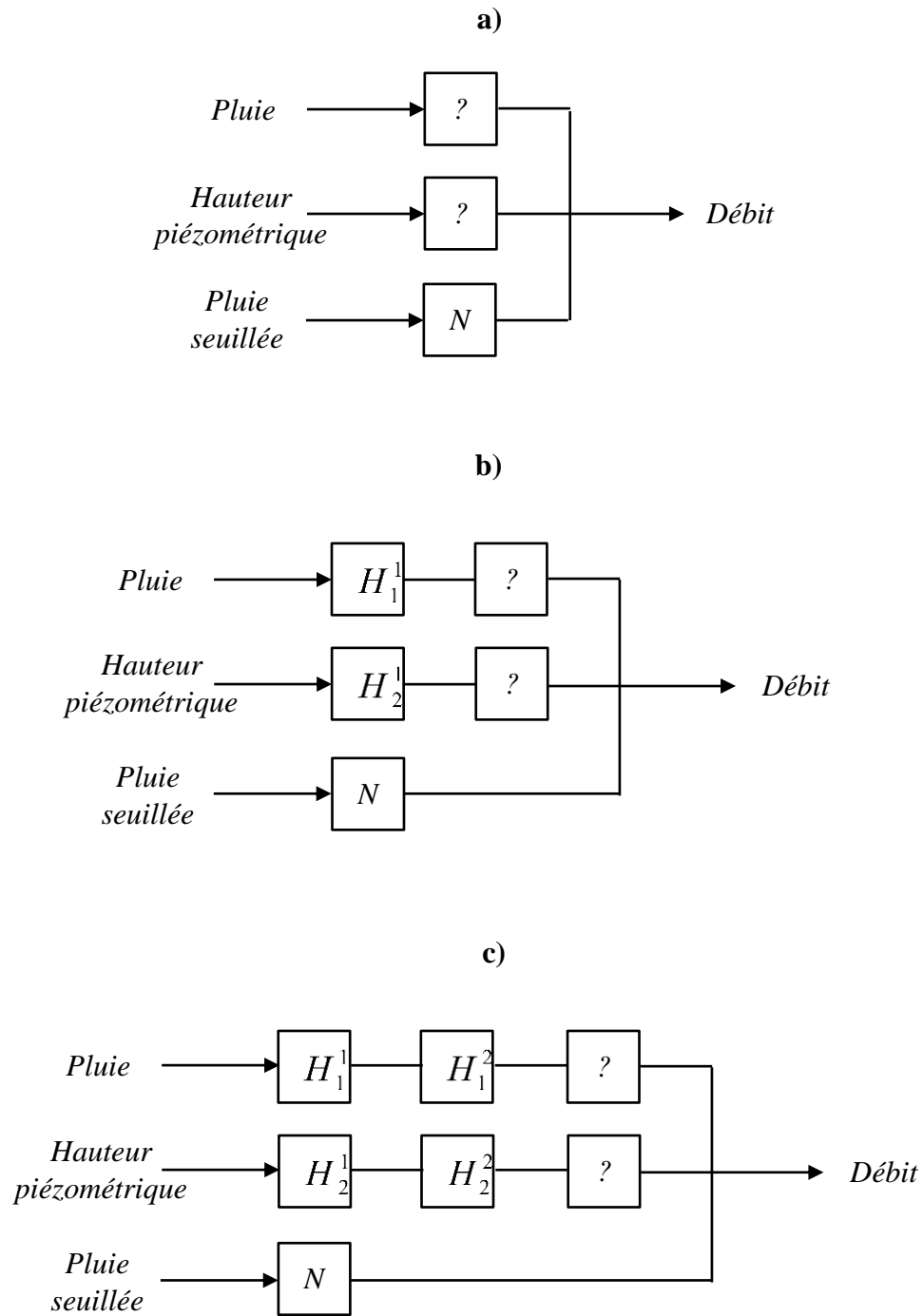


Figure IV-16 : Recherche de la meilleure fonction de transfert pour un modèle à trois branches et (a) une, (b) deux et (c) trois fonctions de transferts par branche. (N : loi normale)

Validation croisée

Afin d'éprouver la robustesse des différentes architectures, tous les tests sont effectués en validation croisée. Pour cela, deux étapes sont nécessaires : dans un premier temps, les paramètres des fonctions de transfert sont optimisés sur une période donnée (phase de calibration). Dans un second temps, ces mêmes paramètres sont utilisés pour reproduire les sorties des autres périodes (phase de prédiction) et l'ajustement de ces simulations est calculé pour vérifier que les modèles sont toujours adéquats. Les chroniques sont divisées en quatre périodes :

- Période 1 : du 1^{er} août 1993 au 31 juillet 1996 (1096 jours)
- Période 2 : du 1^{er} août 1996 au 31 juillet 1999 (1095 jours)
- Période 3 : du 1^{er} août 1999 au 31 juillet 2002 (1096 jours)
- Période 4 : du 1^{er} août 2002 au 31 juillet 2005 (1096 jours)

Les résultats sont représentés de manière concise dans des matrices où le coefficient $N(i,j)$ est l'ajustement mesuré pour les données de la période i , avec des paramètres calculés avec les données de la période j . Les coefficients de la diagonale représentent donc l'ajustement obtenu en simple calibration et non en prédiction. Le fait de travailler en validation croisée permet d'évaluer la performance des modèles, en mesurant leur capacité à simuler les sorties de périodes pour lesquelles ils n'ont pas été calibrés. Ils seront ainsi performants pour effectuer des prévisions, sous réserve que le système modélisé reste invariant. Cela permet notamment de s'assurer que les modèles ne sont pas affectés par le sur-ajustement (*overfitting*). Il arrive en effet que certains modèles mathématiques soient extrêmement bien ajustés sur les périodes de calage mais très mal ajustés sur les autres périodes [Andréassian et al., 2012]. Les modèles sur-ajustés décrivent alors un bruit ou une erreur aléatoire, plutôt que la relation. Dans l'idéal, un modèle parfaitement robuste devrait être calé sur une période la plus courte possible, et donner des simulations de qualité sur la période la plus longue possible. Cela implique d'avoir un modèle suffisamment complexe pour prendre en compte tous les phénomènes existants, mais suffisamment simple pour ne pas surinterpréter les données.

IV.3.2 Résultats

Dans les trois premiers tableaux ci-après (**Tableau IV-3** à Erreur ! Source du renvoi introuvable.) sont reportés les modèles ayant obtenus le meilleur ajustement lors de la phase de calibration. Les termes diagonaux de ces matrices sont égaux aux ajustements en calibration. Les termes non-diagonaux sont égaux aux ajustements obtenus en prédiction : les paramètres du modèle sont calculés sur la période indiquée en ordonnée et l'ajustement du modèle est calculé sur la période indiquée en abscisse, en utilisant ces mêmes paramètres. Ces tableaux permettent de visualiser si les modèles sont performants.

Les deux tableaux suivants (**Tableau IV-6** et **Tableau IV-7**) contiennent les données relatives aux modèles les mieux ajustés respectivement en calibration et en prédiction. Ces deux tableaux permettent de voir rapidement quelles sont les architectures donnant les meilleurs résultats.

Enfin, les **Figure IV-17** à **Figure IV-19** représentent les courbes de débits observés et simulés en utilisant les meilleurs modèles, selon l'architecture de modèle utilisée.

Tableau IV-3 : Matrice des ajustements (coefficients de Nash-Sutcliffe) en validation croisée des meilleures architectures pour un modèle à une branche et (a) une FT par branche (b) deux FT par branche (c) trois FT par branche.

C : période complète (1993 – 2005) — **P_i** : période *i* — **G** : loi Gamma — **N** : loi normale — **B** : loi Béta —
P : loi puissance — Fonctions de transfert en série avec les initiales juxtaposées (exemple : PN = loi puissance et loi normale en série) — Fonctions de transfert en parallèle avec les initiales séparées par un tiret (exemple : G-P = loi Gamma et loi puissance en parallèle).

a)

		Calibration				
		C	P1	P2	P3	P4
Prédiction	C	0.734	0.729	0.691	0.724	0.721
	P1	0.786	0.790	0.734	0.768	0.780
	P2	0.674	0.660	0.710	0.657	0.630
	P3	0.657	0.641	0.591	0.667	0.638
	P4	0.768	0.772	0.682	0.749	0.783
		P	P	P	P	P

b)

		Calibration				
		C	P1	P2	P3	P4
Prédiction	C	0.785	0.775	0.732	0.775	0.773
	P1	0.834	0.837	0.766	0.817	0.830
	P2	0.696	0.679	0.741	0.670	0.654
	P3	0.736	0.707	0.653	0.753	0.716
	P4	0.824	0.822	0.734	0.801	0.843
		PG	PN	PG	PG	PG

c)

		Calibration				
		C	P1	P2	P3	P4
Prédiction	C	0.786	0.781	0.737	0.777	0.773
	P1	0.835	0.841	0.772	0.821	0.830
	P2	0.704	0.678	0.743	0.672	0.653
	P3	0.733	0.721	0.662	0.753	0.718
	P4	0.827	0.829	0.731	0.805	0.843
		PGB	PNG	PGB	PGN	PGP

Tableau IV-4 : Matrice des ajustements (coefficients de Nash-Sutcliffe) en validation croisée des meilleures architectures pour un modèle à deux branches et (a) une FT par branche (b) deux FT par branche (c) trois FT par branche.

C : période complète (1993 – 2005) — P_i : période i — G : loi Gamma — N : loi normale — B : loi Béta — P : loi puissance — Fonctions de transfert en série avec les initiales juxtaposées (exemple : PN = loi puissance et loi normale en série) — Fonctions de transfert en parallèle avec les initiales séparées par un tiret (exemple : G-P = loi Gamma et loi puissance en parallèle).

a)

		Calibration				
		C	P1	P2	P3	P4
Prédiction	C	0.828	0.809	0.801	0.793	0.810
	P1	0.836	0.848	0.809	0.794	0.822
	P2	0.812	0.759	0.845	0.709	0.769
	P3	0.801	0.758	0.750	0.817	0.775
	P4	0.850	0.834	0.793	0.812	0.866
		G-P	G-N	G-P	G-B	G-P

b)

		Calibration				
		C	P1	P2	P3	P4
Prédiction	C	0.830	0.812	0.801	0.811	0.815
	P1	0.842	0.853	0.809	0.814	0.831
	P2	0.796	0.722	0.845	0.734	0.768
	P3	0.810	0.778	0.750	0.831	0.779
	P4	0.854	0.845	0.793	0.817	0.869
		GP-PG	GP-NB	GP-PP	GB-BG	GB-PN

c)

		Calibration				
		C	P1	P2	P3	P4
Prédiction	C	0.829	0.813	0.802	0.810	0.815
	P1	0.843	0.853	0.809	0.814	0.833
	P2	0.804	0.722	0.845	0.731	0.765
	P3	0.800	0.779	0.751	0.832	0.780
	P4	0.852	0.848	0.793	0.815	0.867
		GPP-PGP	GPN-NBB	GPN-PPP	GBB-BGB	GBN-PNP

Tableau IV-5 : Matrice des ajustements (coefficients de Nash-Sutcliffe) en validation croisée des meilleures architectures pour un modèle à trois branches et (a) une FT par branche (b) deux FT par branche (c) trois FT par branche.

C : période complète (1993 – 2005) — P_i : période i — G : loi Gamma — N : loi normale — B : loi Béta — P : loi puissance — Fonctions de transfert en série avec les initiales juxtaposées (exemple : PN = loi puissance et loi normale en série) — Fonctions de transfert en parallèle avec les initiales séparées par un tiret (exemple : G-P = loi Gamma et loi puissance en parallèle).

a)

		Calibration				
		C	P1	P2	P3	P4
Prédiction	C	0.845	0.780	0.811	0.804	0.768
	P1	0.898	0.946	0.849	0.817	0.805
	P2	0.831	0.778	0.856	0.717	0.693
	P3	0.776	0.578	0.730	0.823	0.663
	P4	0.832	0.688	0.777	0.806	0.918
		G-P-N	G-P-N	G-P-N	G-G-N	G-P-N

b)

		Calibration				
		C	P1	P2	P3	P4
Prédiction	C	0.849	0.787	0.811	0.811	0.763
	P1	0.902	0.949	0.849	0.819	0.801
	P2	0.816	0.758	0.856	0.728	0.684
	P3	0.790	0.600	0.730	0.834	0.658
	P4	0.837	0.706	0.777	0.807	0.917
		GP-PN-N	GB-PP-N	GP-PP-N	GB-GB-N	GN-PP-N

c)

		Calibration				
		C	P1	P2	P3	P4
Prédiction	C	0.848	0.783	0.814	0.813	0.759
	P1	0.902	0.948	0.854	0.817	0.800
	P2	0.815	0.731	0.856	0.739	0.663
	P3	0.789	0.607	0.733	0.832	0.664
	P4	0.837	0.705	0.777	0.820	0.906
		GPP-PNN-N	GBP-PPG-N	GPB-PPP-N	GBB-GBB-N	GNN-PPG-N

Tableau IV-6 : Ajustements (coefficients de Nash-Sutcliffe) des meilleurs modèles en calibration pour les architectures à (a) une branche (b) deux branches (c) trois branches.

C : période complète (1993 – 2005) — **P_i** : période *i* — **G** : loi Gamma — **N** : loi normale — **B** : loi Béta —
P : loi puissance — Fonctions de transfert en série avec les initiales juxtaposées (exemple : PN = loi puissance et loi normale en série) — Fonctions de transfert en parallèle avec les initiales séparées par un tiret (exemple : G-P = loi Gamma et loi puissance en parallèle).

Nombre de FT	a)			b)		
	1	2	3	1	2	3
C	P	PG	PGB	G-P	GP-PG	GPP-PGP
	0.734	0.785	0.786	0.828	0.830	0.829
P1	P	PN	PNG	G-N	GP-NB	GPN-NBB
	0.790	0.837	0.841	0.848	0.853	0.853
P2	P	PG	PGB	G-P	GP-PP	GPN-PPP
	0.710	0.741	0.743	0.845	0.845	0.845
P3	P	PG	PGN	G-B	GB-BG	GBB-BGB
	0.667	0.753	0.753	0.817	0.831	0.832
P4	P	PG	PGP	G-P	GB-PN	GBN-PNP
	0.783	0.843	0.843	0.866	0.869	0.867

c)

Nombre de FT	1	2	3
C	G-G-N	GP-PN-N	GPP-PNN-N
	0.844	0.849	0.848
P1	G-P-N	GB-PP-N	GBP-PPG-N
	0.946	0.949	0.948
P2	G-P-N	GP-PP-N	GPB-PPP-N
	0.856	0.856	0.856
P3	G-G-N	GB-GB-N	GBB-GBB-N
	0.823	0.834	0.832
P4	G-P-N	GN-PP-N	GNN-PPG-N
	0.918	0.917	0.906

Tableau IV-7 : Ajustements (coefficients de Nash-Sutcliffe) des meilleurs modèles en validation (les paramètres sont calibrés sur la période complète) pour les architectures à (a) une branche (b) deux branches (c) trois branches.

P_i : période *i* — G : loi Gamma — N : loi normale — B : loi Béta — P : loi puissance — Fonctions de transfert en série avec les initiales juxtaposées (exemple : PN = loi puissance et loi normale en série) — Fonctions de transfert en parallèle avec les initiales séparées par un tiret (exemple : G-P = loi Gamma et loi puissance en parallèle).

Nombre de FT	a)			b)		
	1	2	3	1	2	3
P1	P	PN	PNG	G-N	GP-NB	GPN-NBB
	0.729	0.775	0.781	0.809	0.812	0.813
P2	P	PG	PGB	G-P	GP-PP	GPN-PPP
	0.691	0.732	0.737	0.801	0.801	0.802
P3	P	PG	PGN	G-B	GB-BG	GBB-BGB
	0.724	0.775	0.777	0.793	0.811	0.810
P4	P	PG	PGP	G-P	GB-PN	GBN-PNP
	0.721	0.773	0.773	0.810	0.815	0.815

c)

Nombre de FT	1	2	3
P1	G-P-N	GB-PP-N	GBP-PPG-N
	0.780	0.787	0.783
P2	G-P-N	GP-PP-N	GPB-PPP-N
	0.811	0.811	0.814
P3	G-G-N	GB-GB-N	GBB-GBB-N
	0.804	0.811	0.813
P4	G-P-N	GN-PP-N	GNN-PPG-N
	0.768	0.763	0.759

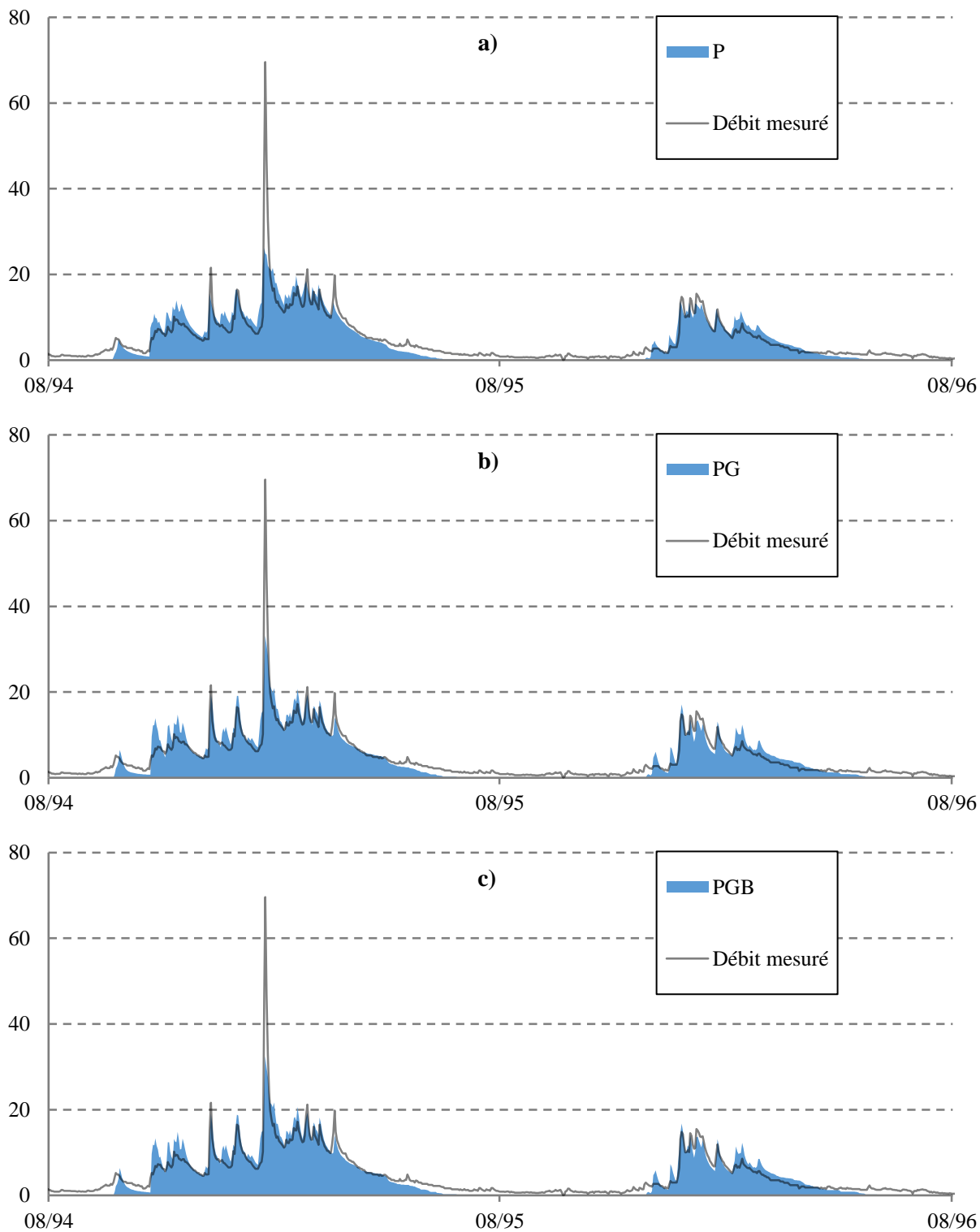


Figure IV-17 : Débit mesuré (courbe en trait plein) et débit simulé (aire) calculé avec les fonctions de transfert donnant le meilleur ajustement en calibration sur la période complète, pour un modèle à une branche et (a) une FT par branche (b) deux FT par branche (c) trois FT par branche. Seules deux années de données sont représentées graphiquement afin de faciliter la lecture des courbes.

P : loi puissance — G : loi Gamma — B : loi Béta — Fonctions de transfert en série avec les initiales juxtaposées (exemple : PG = loi puissance et loi Gamma en série).

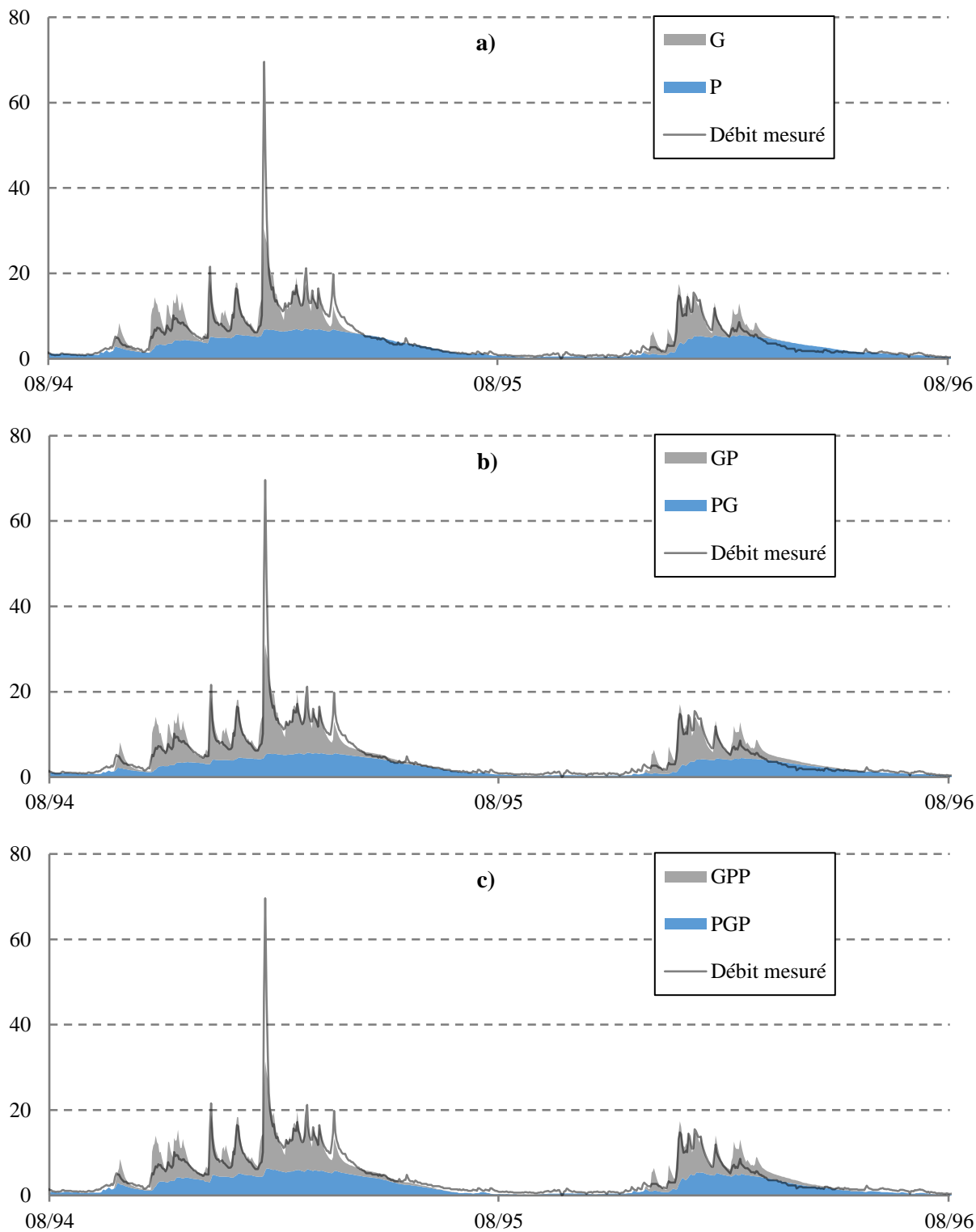


Figure IV-18 : Débit mesuré (courbe en trait plein) et débits simulés (aires empilées) calculés avec les fonctions de transfert donnant le meilleur ajustement en calibration sur la période complète, pour un modèle à deux branches et (a) une FT par branche (b) deux FT par branche (c) trois FT par branche. Seules deux années de données sont représentées graphiquement afin de faciliter la lecture des courbes.

P : loi puissance — G : loi Gamma — Fonctions de transfert en série avec les initiales juxtaposées (exemple : PG = loi puissance et loi Gamma en série).

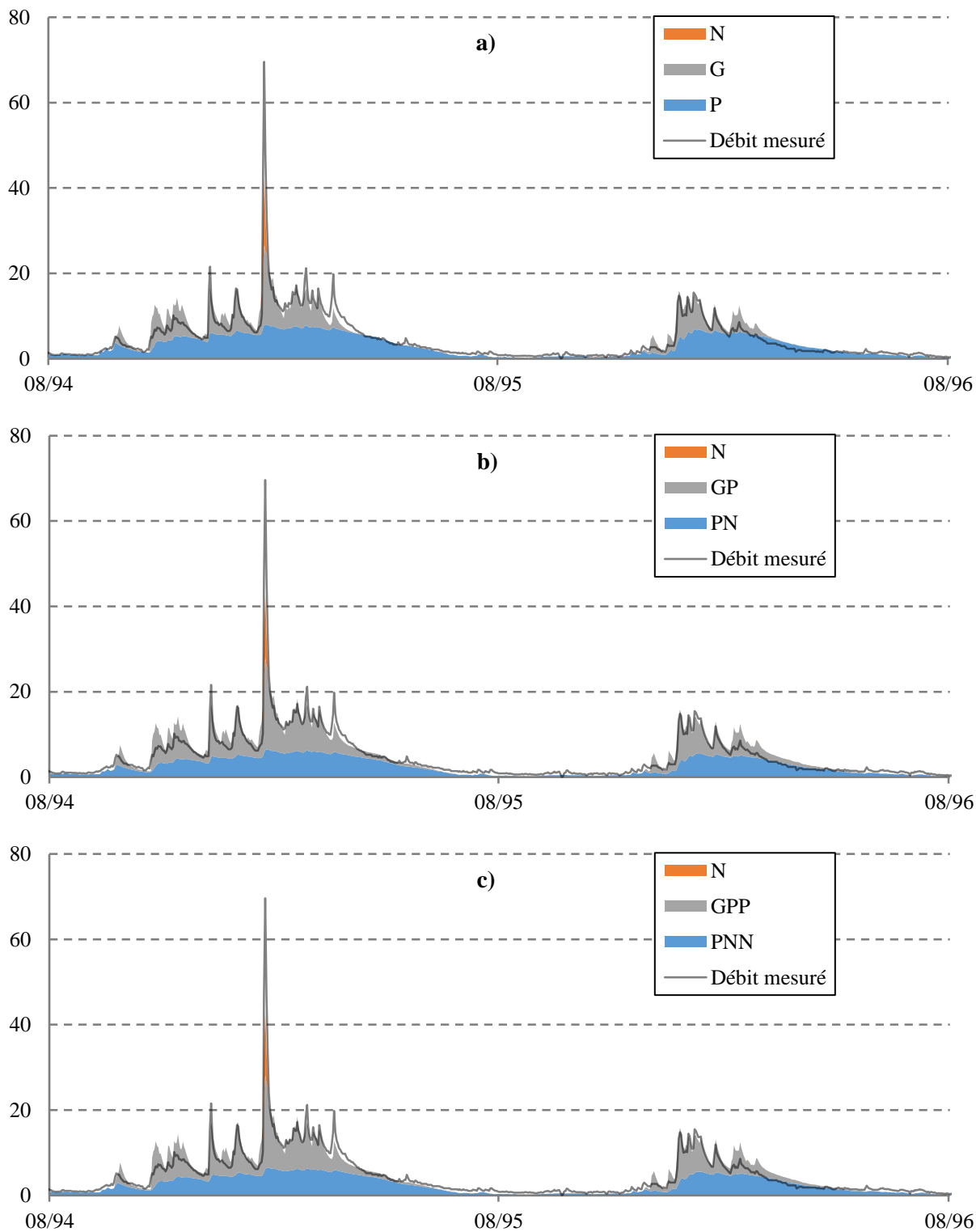


Figure IV-19 : Débit mesuré (courbe en trait plein) et débits simulés (aires empilées) calculés avec les fonctions de transfert donnant le meilleur ajustement en calibration sur la période complète, pour un modèle à trois branches et (a) une FT par branche (b) deux FT par branche (c) trois FT par branche. Seules deux années de données sont représentées graphiquement afin de faciliter la lecture des courbes.

P : loi puissance — G : loi Gamma — N : loi normale — Fonctions de transfert en série avec les initiales juxtaposées (exemple : PG = loi puissance et loi Gamma en série).

IV.3.3 Discussion des résultats

Les architectures présentées dans les tableaux et figures ci-dessus (**Tableau IV-3** à **Tableau IV-7** et **Figure IV-17** à **Figure IV-19**) sont celles qui donnent le meilleur ajustement en calibration. D'autres résultats sur des architectures moins performantes sont reportés en annexe (**Tableau B-1** à **Tableau B-5**).

Pour commencer, il est intéressant d'analyser les types de fonction de transfert ayant donné les meilleurs ajustements pour le modèle à une branche. Pour le modèle le plus simple à une fonction de transfert, où l'entrée est la pluie nette, la fonction en loi puissance est la meilleure, quelle que soit la période utilisée pour calibrer ses paramètres (**Tableau IV-3, a**). Cette fonction définie par une trainée importante (c'est-à-dire des valeurs non nulles aux temps longs) permet de bien reproduire les périodes d'étiages (**Figure IV-20, a**). Quand une seconde fonction en série est ajoutée à ce modèle à une branche, les meilleures performances sont observées pour la loi Gamma (pour quatre périodes de calibration sur cinq), suivie de la loi normale (**Tableau IV-3, b**). Ces deux fonctions vont apporter deux modifications importantes à la forme de la fonction de transfert équivalente (**Figure IV-20, b**) :

1. Le maximum de la fonction équivalente est décalé vers les temps positifs. Pour la sortie, cela a pour effet de décaler les débits de pic consécutifs aux épisodes pluvieux de quelques jours.
2. La courbure de la fonction équivalente est accentuée (i.e. la valeur absolue de sa dérivée augmente). Au niveau de la sortie, les courbes de récession sont donc plus courtes.

En moyenne, les ajustements sont améliorés de 7,6% en calibration et de 6,7% en prédiction en passant d'une à deux FT en série. Ces pourcentages sont les taux de variation de l'ajustement entre les modèles possédant une et deux FT. Il est intéressant de noter que l'algorithme a donc choisi dans un premier temps de bien simuler les temps longs (pour le modèle à une FT), puis dans un second temps les temps courts (pour le modèle à deux FT). Ce comportement était néanmoins attendu, étant donné que les périodes de récessions sont les plus représentées dans les données et pèsent donc plus dans la fonction objectif, même si cette dernière a tendance à plus pénaliser les erreurs portant sur les fortes valeurs. La troisième fonction de transfert est différente pour pratiquement chaque période de calibration : loi Béta, loi Gamma, loi normale ou loi puissance (**Tableau IV-3, c**). La forme de la fonction de transfert équivalente est peu impactée par cette troisième fonction de transfert, seul le pic de la courbe est légèrement modifié (**Figure IV-20, c**). En moyenne, les ajustements sont améliorés de 0,2% en calibration et de 0,4% en prédiction en passant de deux à trois FT en série. En conclusion, l'architecture à une fonction de transfert donne des résultats satisfaisants, l'ajout d'une seconde FT améliore significativement le modèle, et l'ajout d'une troisième FT n'apporte pas spécialement d'améliorations.

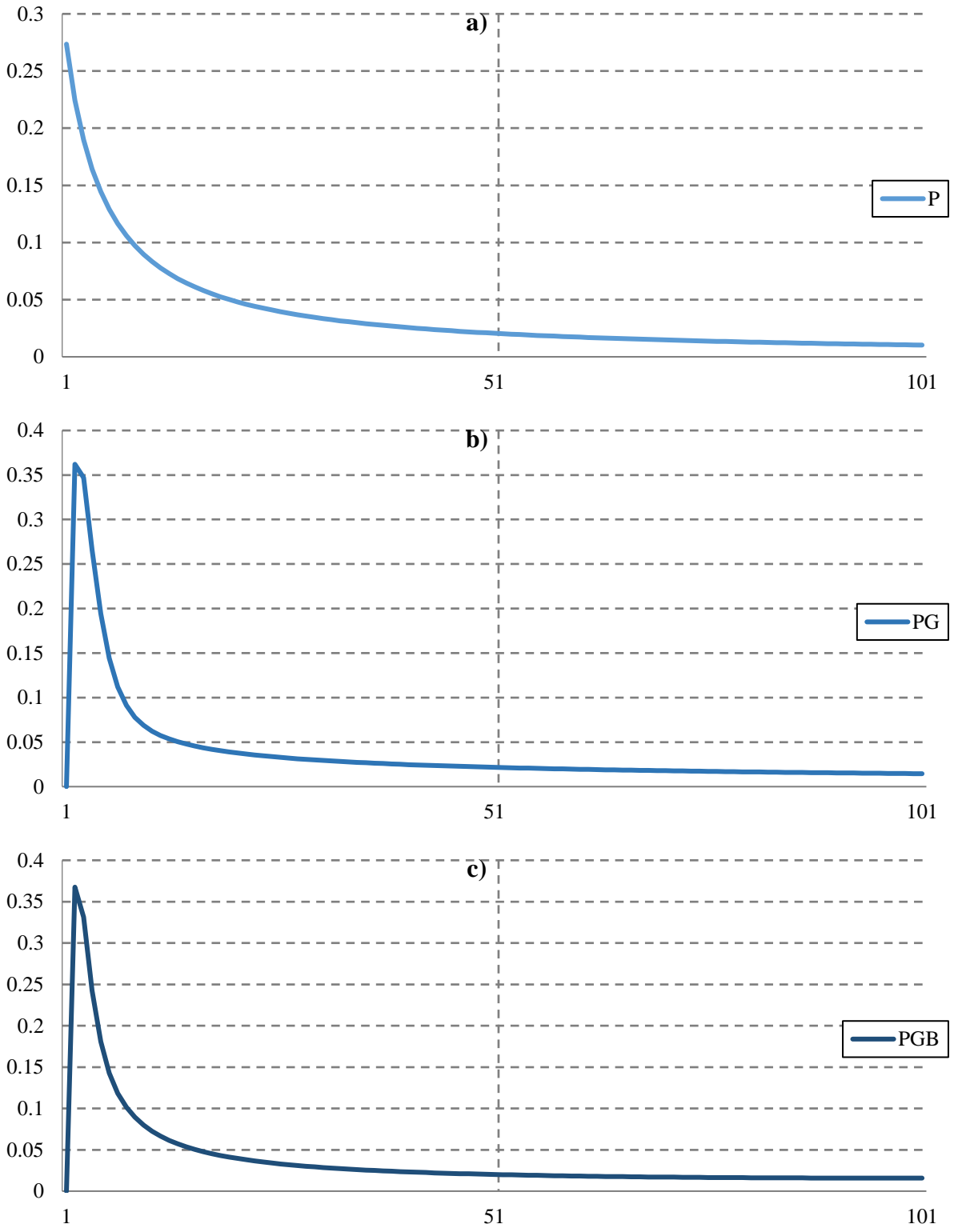


Figure IV-20 : Fonctions de transfert équivalentes

P : loi puissance — G : loi Gamma — B : loi Béta — Fonctions de transfert en série avec les initiales juxtaposées (exemple : PG = loi puissance et loi Gamma en série)

Les modèles comportant deux branches utilisent deux chroniques en entrée : la pluie nette pour la première, la hauteur piézométrique pour la seconde

Le fait d'ajouter cette deuxième entrée apportant beaucoup d'information améliore nettement les simulations. En moyenne, les ajustements augmentent de 13,8%, 7,0% et 6,5% par rapport au modèle à une branche comportant respectivement une, deux et trois fonctions de transfert par branche. Pour le modèle à une fonction de transfert par branche, la fonction utilisée pour convoluer la pluie n'est plus la loi puissance mais la loi Gamma (Tableau 4, a). En effet, la reproduction des débits d'étiage aux temps longs est assurée par la branche utilisant la piézométrie en entrée qui utilise indifféremment une loi puissance, normale ou Béta selon les périodes de calibration considérées. La principale raison de ce changement est que le signal piézométrique a de l'inertie ce qui fait que sa forme est plus proche de la forme du débit en récession. De plus, lors des phases de récession, une bonne partie du débit observé est probablement le fait du débit naturel de la nappe du Dogger, étant donné l'importance de la karstification de cette dernière. Les temps longs et courts étant correctement simulés en utilisant une fonction par branche, les modèles plus complexes n'apportent que peu ou pas d'améliorations : 0,6% en moyenne entre les modèles à une et deux fonctions par branche ; - 0,03% entre les modèles à deux et trois fonctions par branche (Tableau IV-4, b et c). Les fonctions supplémentaires sont donc de toutes les formes, indifféremment loi Gamma, Béta, normale ou puissance et leur poids ne modifie pratiquement pas la forme de la fonction équivalente.

Les modèles à trois branches prennent en compte la pluie seuillée comme entrée supplémentaire. Cette variable a été créée pour améliorer l'ajustement des fonctions de transfert en calibration. Ce rôle est rempli car par rapport aux modèles à deux branches, les modèles à trois branches améliorent en moyenne l'ajustement en calibration de 4,3%, 4,1% et 3,8% pour respectivement une, deux et trois fonctions par branche (**Erreur ! Source du renvoi introuvable.**). En revanche, en prédiction, les performances moyennes baissent respectivement de -1,7%, -1,9% et -2,0%. Ce modèle est donc sur-ajusté car l'amélioration de l'ajustement en calibration s'est fait au détriment du caractère général du modèle en détériorant l'ajustement en prédiction. Pour ce qui est du type de fonction de transfert, la seule tendance qui se dégage est l'utilisation systématique de la fonction Gamma pour convoluer la pluie nette, et quasi-systématique de la fonction loi puissance pour la hauteur piézométrique pour le modèle à une fonction par branche.

En conclusion, pour les différentes architectures MASH testées dans cet exemple :

1. Type de fonctions de transfert

Deux fonctions se démarquent des autres fonctions de transfert : les lois puissance et Gamma. L'avantage de la loi puissance est sa décroissance sur de longues périodes de temps, permettant de simuler efficacement les débits d'étiages. L'intérêt de la loi Gamma est le décalage de son pic vers les temps courts, associé à la décroissance prononcée au-delà (temps longs) du pic.

2. Fonctions en série

Pour les modèles à deux ou trois branches, une fonction de transfert par branche suffit, et le fait de rajouter d'autres fonctions en série ne fait qu'agrandir l'espace des solutions et donc complexifier l'optimisation des paramètres sans améliorer la qualité de l'ajustement (**Tableau IV-6** et **Tableau IV-7**). En revanche, les modèles à une branche sont plus performants avec deux fonctions en série plutôt qu'une car ces deux fonctions remplissent des rôles différents, l'une simulant les temps courts, l'autre les temps longs. Cela est vrai en calibration aussi bien qu'en prédiction.

3. Fonctions en parallèles

En calibration, la mise de fonction en parallèle améliore systématiquement la qualité de l'ajustement. Cela provient du fait que la mise en parallèle va de pair avec l'ajout de nouvelles entrées, ce qui apporte plus de données à même d'expliquer la sortie. En prédiction, le modèle à deux branches est plus performant que le modèle à une branche, car ici aussi une branche va simuler les temps courts et l'autre les temps longs. L'ajout d'une troisième branche a un impact plus mitigé.

4. Calibration et prédiction

La limite à l'ajout de nouvelles entrées et à la complexification du modèle est le sur-apprentissage du modèle. En effet, certaines entrées, si elles vont permettre d'améliorer l'ajustement du modèle en calibration, vont créer des fonctions qui représenteront moins bien le système dans sa globalité et fourniront des ajustements suboptimaux en prédiction sur d'autres périodes. C'est exactement ce qui se passe avec les modèles recevant la pluie seuillée en troisième entrée, et dont les ajustements sont soit semblables, soit inférieurs aux ajustements des modèles plus simples à deux entrées (**Tableau IV-6** et **Tableau IV-7**). Dans cet exemple, le meilleur modèle est donc le modèle à deux branches et deux fonctions par branche, pour sa robustesse, sa simplicité et la qualité des ajustements qu'il fournit.

Ces conclusions ne sont évidemment valides que pour le cas d'étude présenté ici. Même si d'autres études sont nécessaires pour s'assurer que le fonctionnement du modèle reste cohérent, cet exemple d'application, par sa complexité, prouve que MASH est prometteur pour la modélisation et la simulation d'hydrosystèmes sophistiqués. L'hydrosystème présenté ici se prête particulièrement bien à la mise en parallèle et en série de fonctions de transfert. Il est en effet caractérisé par une réponse rapide, dû au système karstique, associée à une réponse lente provenant du fait que la Sèvre Niortaise soit l'exutoire de la nappe du Dogger ainsi que de sous-bassins qui constituent autant de systèmes parallèles.

CONCLUSION GÉNÉRALE

En classant les modèles hydrogéologiques selon la manière dont ils représentent les processus, trois catégories principales se distinguent : les modèles à base physique, les modèles conceptuels et les modèles systémiques. Les modèles à base physique découpent l'espace en mailles dans lesquelles des lois mathématiques décrivent les processus physiques majeurs sensés reproduire les observations avec un degré d'incertitude caractérisable. Les modèles conceptuels simulent le fonctionnement macroscopique du système grâce à des lois mathématiques relativement simples, décrivant un ensemble de réservoirs interconnectés. Les modèles systémiques utilisent des méthodes purement numériques afin de recréer un signal de sortie à partir d'un signal d'entrée, ignorant le plus souvent les processus physiques au profit du meilleur ajustement possible.

Un modèle systémique nommé MASH (Modélisation des Hydrosystèmes par Approche Systémique) a été développé dans ce travail, en essayant de limiter certains défauts des modèles systémiques (notamment leur absence de prise en compte des variabilités spatiales et leur manque d'information déduite du modèle sur le fonctionnement de l'hydrosystème) tout en gardant certains de leurs avantages (leur simplicité, leur parcimonie et leur besoin plus limité en données). Pour ce faire, MASH possède deux particularités : 1- le modèle est basé sur l'utilisation de fonctions de transfert (ou noyaux de convolution) paramétriques. Ces fonctions de transfert ont été choisies pour leur faible paramétrisation, leur caractère général et le sens physique des mécanismes qu'elles miment ou simulent. Quatre fonctions sont pour l'instant disponibles, toutes étant des fonctions de densité de probabilité : la loi Gamma, la loi normale, la loi Bêta et la loi exponentielle. 2- le modèle est versatile, au sens où son architecture est modulable et que le nombre d'entrées, le nombre de fonctions de transfert en série et le type de fonctions de transfert utilisé sont laissés à la discrétion de l'utilisateur. Le nombre d'entrées, compris entre un et quatre, définit également le nombre de branches parallèles que comporte le modèle. Le nombre de fonctions de transfert en série, compris entre un et trois, impose la taille de chacune des branches. Le type de fonction de transfert paramétrique est choisi pour chacune des fonctions du modèle au sein d'une bibliothèque comportant les quatre fonctions décrites précédemment. MASH possède donc dans sa version la plus simple une entrée et une fonction de transfert, et dans sa version la plus complexe quatre entrées et douze fonctions de transfert. Des développements ultérieurs pourront amender ce modèle, d'une part en ajoutant de nouvelles fonctions de transfert à la bibliothèque de fonctions déjà disponible, et d'autre part en modifiant son architecture, c'est-à-dire le nombre de fonctions en parallèle et en série, afin de répondre à des problèmes hydrogéologiques spécifiques. Cela étant, augmenter le nombre de branches et le nombre de noyaux en série ne va pas dans le sens de la philosophie "parcimonieuse" des modèles systémiques. Il peut sembler tout aussi intéressant de travailler sur des préprocesseurs des entrées et des sorties afin de mieux cerner des caractéristiques spécifiques d'un bassin versant (par exemple, des écoulements hypodermiques systématiques sur un bassin agricole où travailler sur l'intrant pluvial et un débit de rivière en récession s'avère plus judicieux pour comprendre le comportement de nappe).

La seconde partie de ce travail s'est efforcée à tester le modèle dans deux configurations : avec des données provenant d'un cas d'école et avec des données provenant d'un hydrosystème réel. Le test sur le cas d'école a eu pour objectif de prouver que MASH était capable de reproduire le comportement de systèmes de complexité croissante et de mettre à l'épreuve sa capacité à identifier et à séparer les contributions au débit provenant de différentes zones à l'aide de fonctions de transfert disposées en parallèle. Afin de disposer d'un cas synthétique dont tous les paramètres sont maîtrisés, plusieurs simulations sont réalisées à l'aide d'un modèle distribué à base physique (code MARTHE développé par le BRGM). À l'issue de cette première batterie de tests, MASH a démontré sa capacité à reproduire efficacement les débits générés par MARTHE, pour des modèles d'une à quatre zones de sol, quels que soient les paramètres de ces différentes zones. De plus, quand le modèle à base physique est constitué de deux zones, les deux fonctions de transfert de MASH représentent de manière univoque ces différentes zones (sauf lorsque celles-ci sont trop peu différenciées ET qu'elles représentent des systèmes à forte inertie). Cependant, quand le modèle MARTHE est constitué de trois ou quatre zones, les trois ou quatre fonctions de transfert de MASH ne correspondent pas forcément à ces différentes zones (sauf lorsque celles-ci représentent des systèmes à faible inertie). Ce manque de différenciation, même s'il n'est pas dérangeant d'un point de vue simulation, car le modèle est toujours capable de bien reproduire les débits, montre les limites de l'approche systémique quand il s'agit de donner un sens physique aux différents réservoirs représentés par les fonctions de transfert du modèle. Cela étant, inverser un modèle distribué hétérogène sur la simple base des chroniques locales en un exutoire unique conduirait également à se méprendre sur l'identification des hétérogénéités vues par le modèle. Concernant MASH, des contraintes supplémentaires sur les paramètres des fonctions de transfert permettraient peut-être d'améliorer ce dernier point, sachant alors qui faudrait disposer sur les sous-systèmes d'un bassin versant d'une information pertinente sur les mécanismes prédominants à l'origine des flux hydriques (ou de ce qui est la cible du modèle).

Une autre piste testée est l'inversion du modèle, cet aspect étant au cœur du problème de modélisation. Plusieurs membres d'une famille d'heuristiques basée sur l'intelligence en essaim nommée « optimisation par essaim de particule » (ou PSO pour « *Particle Swarm Optimization* ») ont été implémentées afin d'étudier leur influence sur les capacités de simulation des modèles systémiques construits. Trois membres de cette famille d'heuristiques ont été testées sur un hydrosystème réel au fonctionnement complexe, le bassin de la Sèvre Niortaise, composé d'aquifères faillés et multicouches, avec de nombreux échanges entre les nappes et le réseau superficiel. Ces trois algorithmes (CR-PSO pour *centered-regressive PSO*, CP-PSO pour *centered-progressive PSO* et CC-PSO pour *centered-centered PSO*) se distinguent quant à leur capacité d'exploration de l'espace des solutions, ou au contraire de convergence rapide vers une ou plusieurs destinations. Pour l'exemple étudié, comprenant 12 années de chroniques ininterrompues de pluie, ETP, hauteur piézométrique et débit à l'exutoire, l'algorithme CP-PSO s'est avéré trop exploratoire. En faisant de grands sauts, les particules arrivent à explorer une partie très vaste de l'espace des solutions, mais la majorité passe à côté de la vallée contenant le minimum global. À l'inverse, l'algorithme CC-PSO converge trop vite pour ce problème : les essaims peinent à trouver des solutions situées au-delà de leur espace de départ car ils se resserrent rapidement dès qu'un de leur agent trouve une solution intéressante.

L'algorithme CR-PSO est le plus modéré : il est plus convergent que le CP-PSO, mais plus exploratoire que le CC-PSO ce qui permet aux essais d'explorer l'espace des solutions sans rester bloqué dans l'espace de départ et sans se perdre dans des zones peu performantes de l'espace des paramètres. Ainsi, les particules évoluent plus lentement vers le minimum global et continuent de converger pendant de nombreuses itérations, même en fin d'optimisation

Ce second exemple d'utilisation, fondé sur des données réelles, a également permis de vérifier la pertinence de l'architecture et des différentes fonctions de transfert proposées, en testant toutes les figurations possibles en validation croisée (calibration avec un plage de données suivie d'une prédiction sur une autre plage de données). Il en ressort que deux fonctions de transfert se démarquent des autres par rapport à leur capacité de prédiction : les lois puissance et Gamma. L'avantage de la loi puissance est sa décroissance sur de longues périodes de temps, permettant de simuler efficacement les débits d'étiages. L'intérêt de la loi Gamma est le décalage de son pic vers les temps courts, associé à la décroissance prononcée au-delà (temps longs) du pic. Au niveau de l'architecture, placer des fonctions en parallèle améliore systématiquement la qualité de l'ajustement en calibration. Cela provient du fait que le système parallèle va de pair avec l'ajout de nouvelles entrées et apporte plus de données à même d'expliquer la sortie. En prédiction, le modèle à deux branches est plus performant que le modèle à une branche, car une branche va simuler les temps courts et l'autre les temps longs. L'ajout d'une troisième branche semble avoir un impact plus mitigé. Concernant l'architecture en série, pour les modèles à deux ou trois branches, une fonction de transfert par branche suffit, et le fait d'ajouter d'autres fonctions en série ne fait qu'agrandir l'espace des solutions et donc complexifier l'optimisation des paramètres sans améliorer la qualité de l'ajustement. En revanche, les modèles à une branche sont plus performants avec deux fonctions en série plutôt qu'une car ces deux fonctions remplissent des rôles différents, l'une simulant les temps courts, l'autre les temps longs. Cela est vrai en calibration aussi bien qu'en prédiction. La limite à l'ajout de nouvelles entrées et à la complexification du modèle est le sur-apprentissage du modèle. En effet, certaines entrées, si elles permettent d'améliorer l'ajustement du modèle en calibration, peuvent également ajouter dans le modèle des fonctions de transfert qui représenteront moins bien le système dans sa globalité et fourniront des ajustements suboptimaux en prédiction. C'est ce qui a pu se passer lors des expériences ayant utilisées une chronique de pluie seuillée en troisième entrée sur le modèle testé dans ce travail, les ajustements étant alors soit semblables, soit inférieurs aux ajustements des modèles plus simples à deux entrées. Dans l'exemple de bassin versant réel testé, le meilleur modèle est donc un système à deux branches et deux fonctions par branche, pour sa robustesse, sa simplicité et la qualité des ajustements qu'il fournit. Bien évidemment, une étude approfondie doit encore être menée sur d'autres exemples afin de valider ces premières conclusions en reproduisant les résultats qui viennent d'être évoqués.

S'agissant des perspectives de recherche, l'étude menée sur le cas d'école synthétique calculé avec MARTHE mériterait d'être poursuivie afin de définir quelle est la sensibilité de MASH aux différences de propriétés physiques entre sous-zones du système modélisé, et plus précisément quelle est l'amplitude des différences perceptibles par le modèle. Il s'agirait de vérifier si au-delà d'un certain degré de différenciation des propriétés des zones, le modèle se

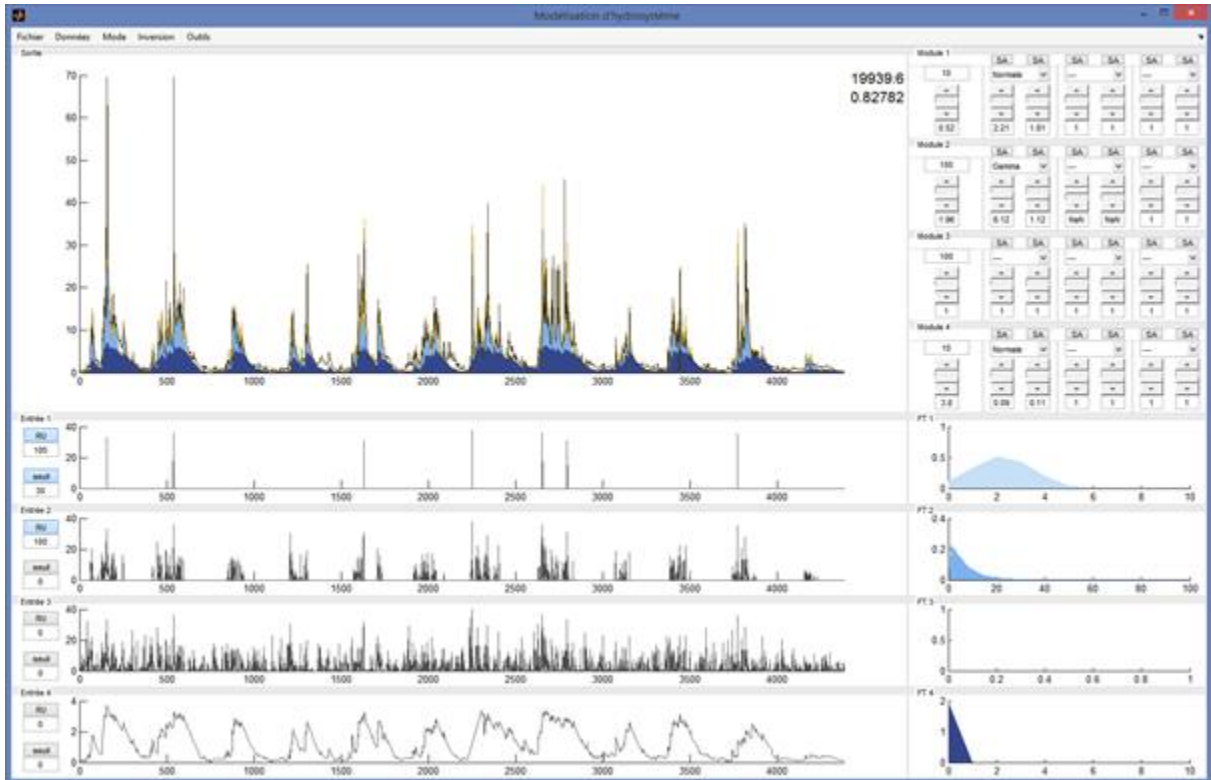
révèle plus performant à détecter chaque zone contributive. Ce point est crucial concernant la problématique de la caractérisation du contenu informatif des données et de la mise en relation entre les données, la physique et le modèle. À ce propos, on notera qu'il serait opportun de mettre à jour une partie de l'étude pour se passer du module GARDENIA et ne garder que le modèle MARTHE, afin de n'utiliser que des paramètres de modèle ayant un sens hydrodynamique, comme la conductivité hydraulique, la transmissivité et le coefficient d'emmagasinement, plutôt que les paramètres (durée de demi-percolation et niveau d'équilibre ruissellement/percolation) utilisés ici. Cela permettrait peut-être d'établir une correspondance plus directe entre les paramètres hydrodynamiques du modèle à base physique et les paramètres des fonctions de transfert du modèle MASH. Dans cette même optique de relier plus précisément les différents éléments de MASH avec la physique, l'expérience menée sur le choix des fonctions de transfert et des architectures devrait être reproduite sur d'autres hydrosystèmes plus simples. Dans ce travail, nous nous sommes concentrés sur un hydrosystème complexe avec l'idée que si le modèle MASH était capable de modéliser correctement les flux d'un tel système, il serait également capable de simuler le comportement de systèmes plus simples. Une étude extensive déterminant quelles fonctions de transfert et quelles architectures donnent les meilleurs résultats sur un ensemble de systèmes archétypaux (bassin urbain, bassin d'évolution de nappes, bassin côtier, etc.) serait une ressource précieuse qui permettrait aux utilisateurs de MASH de savoir, en première approche, comment paramétrer le modèle.

Diverses autres modifications pourraient être apportées au modèle MASH. Citons par exemple le calcul automatique des paramètres liés aux non-linéarités (la réserve utile qui permet de calculer la pluie efficace, ou le paramètre qui seuille les entrées). Ces paramètres sont pour l'instant fixés a priori par l'utilisateur mais pourraient être identifiés par un algorithme d'inversion. Afin de mieux représenter les extrêmes, il faudrait étudier les manières dont on peut introduire plus de non-linéarité dans le modèle, par exemple en proposant d'autres prétraitements pour les chroniques d'entrée. Pour que le modèle soit capable de représenter des tendances ou des événements cycliques, on pourrait aussi envisager de transformer les paramètres scalaires des fonctions de transfert en fonctions paramétriques simples (fonctions linéaires ou trigonométriques), dont les formes algébriques seraient déterminées par des algorithmes d'inversion. À ce sujet, le remplacement de PEST par un algorithme de Levenberg-Marquardt codé directement dans MASH serait le bienvenu afin de limiter les adhérences (dépendances vis-à-vis d'autres outils) du logiciel et faciliter son installation. Pour finir, le modèle MASH mériterait d'être redéveloppé en tant qu'application web utilisable depuis n'importe quel ordinateur connecté à internet et réécrit dans un langage de programmation plus récent et plus ouvert, comme Python, avec un code source sous licence ouverte disponible sur une forge publique.

ANNEXE A : DESCRIPTION DE L'INTERFACE GRAPHIQUE DE MASH

L'interface graphique utilisateur (GUI) du modèle MASH est réalisée à l'aide de MATLAB. Deux options sont possibles pour construire une interface graphique sous MATLAB : l'outil GUIDE (*Graphical User Interface Design Environment*) ou bien la programmation directe des différents éléments composants l'interface. C'est cette deuxième option qui a été retenue car même si l'utilisation de GUIDE facilite grandement la création de l'interface graphique et des éléments qui la constitue, une partie du code créé par l'outil est inaccessible au développeur ce qui est très problématique pour un programme dont on veut maîtriser totalement le fonctionnement. Chaque élément a donc été programmé directement en écrivant le code source pour plus de lisibilité, pour garder le contrôle sur le fonctionnement du code mais aussi pour s'assurer que le code est facilement maintenable. La fenêtre principale de MASH est représentée (**Figure A-1, a**). Elle est elle-même divisée en quatre parties (**Figure A-1, b**) représentant les entrées du modèle, les sorties, les fonctions de transfert et enfin les paramètres qui y sont associés.

a)



b)

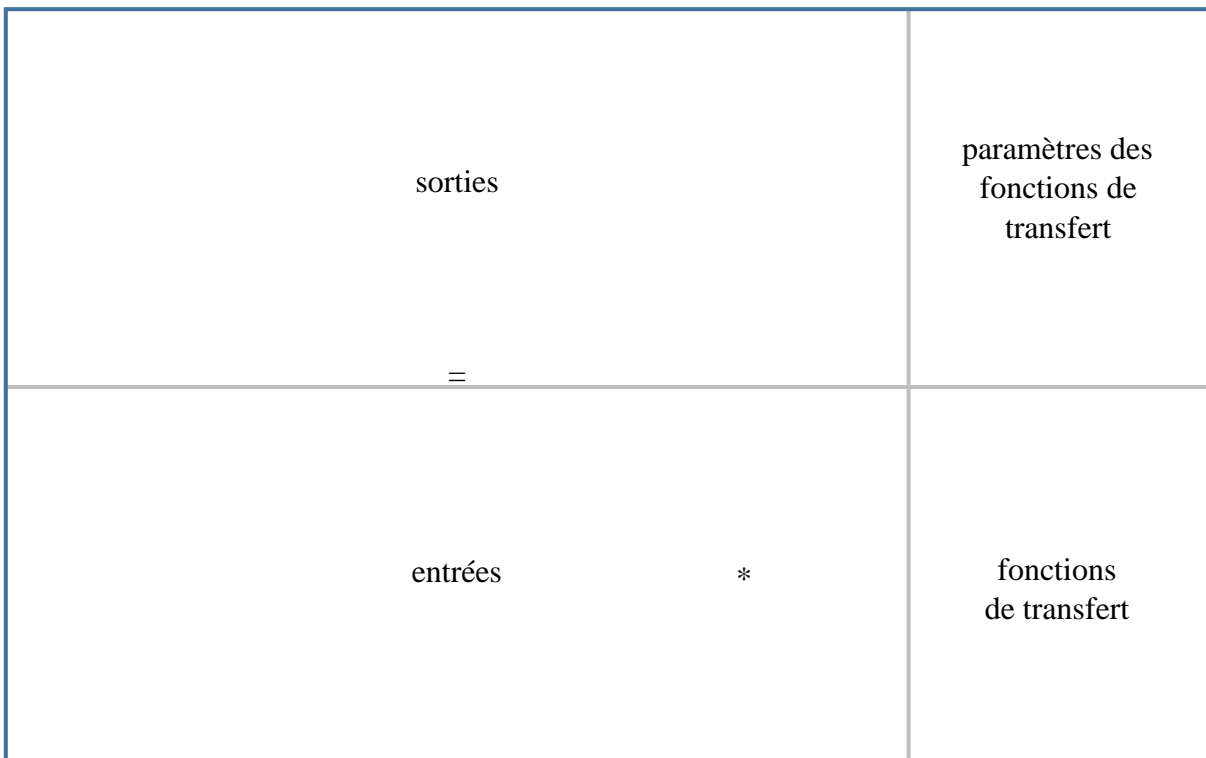


Figure A-1 : Interface graphique utilisateur MASH : (a) capture d'écran ; (b) schéma explicatif correspondant

A.1 Les entrées

La partie inférieure gauche de l'interface graphique réunit les éléments relatifs aux entrées du modèle (Figure A-2). Les quatre entrées du modèle sont tracées sur quatre graphiques superposés. Les axes des abscisses sont identiques de manière à pouvoir étudier la concordance temporelle des variables d'entrée. Si les chroniques d'entrée n'ont pas le même nombre d'élément, c'est le nombre d'élément le plus petit qui est utilisé. Par exemple, si deux chroniques sont importées, 2000 données pour la pluie et 1900 données pour la hauteur piézométrique, les deux chroniques ne sont représentées que jusqu'au pas de temps 1900 (et la sortie n'est pas calculée au-delà).

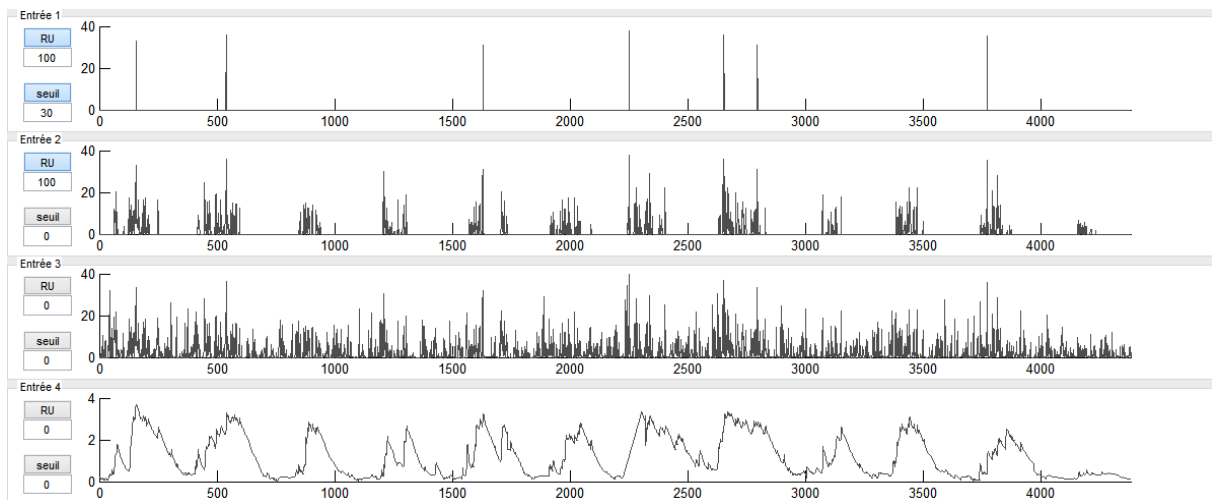


Figure A-2 : Détail de l'interface graphique utilisateur : les entrées

À droite de chaque graphique se trouve deux boutons accompagnés de deux zones de texte modifiable. Le premier bouton « RU » est utilisé pour signaler que la chronique correspondante est une chronique de pluie brute qui doit être transformée en pluie efficace à l'aide d'une chronique d'ETP. En cliquant sur le bouton, MASH demande à l'utilisateur de charger une chronique d'ETP si ce n'est pas déjà fait, puis calcule la pluie efficace grâce la réserve utile maximum R_{\max} défini par l'utilisateur dans la zone de texte (cf. paragraphe II.1.2).

Le deuxième bouton « seuil » permet quant à lui de seuiller une chronique : toutes les valeurs inférieures à la valeur définie dans la zone de texte sont rendues égales à zéro quand ce bouton est activé (cf. paragraphe II.1.3).

A.2 La sortie

Les éléments représentant la sortie sont situés directement au-dessus des entrées (Figure A-3).

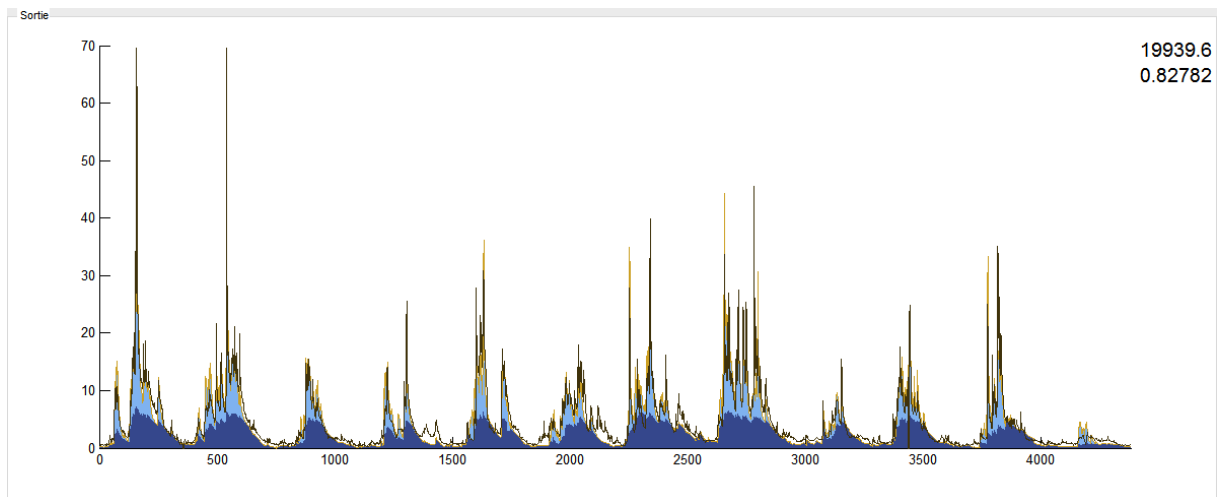


Figure A-3: Détail de l'interface graphique utilisateur : la sortie

La sortie est tracée dans un graphique dont l'axe des abscisses correspond à celui utilisé pour les différentes entrées ce qui permet de relier facilement les événements entrée-sortie ayant lieu simultanément, ou d'observer leur éventuel décalage. Si le nombre de donnée n_s de sortie est inférieur au nombre de données d'entrée n_e , seuls les n_s premier éléments de la chronique d'entrée sont utilisés. Par exemple, dans le cas d'un modèle pluie-débit, s'il y a 5000 données de pluie et 1000 données de débits, seules les 1000 premières données de la chronique de pluie sont utilisées. Dans le cas n_e est supérieur à n_s , un message d'erreur s'affiche, invitant l'utilisateur à choisir une chronique d'entrée au moins aussi longue que le chronique de sortie.

Plusieurs courbes sont tracées sur le graphique :

- La courbe en trait fin de couleur noir correspond à la sortie mesurée. Dans le cadre d'un modèle pluie-débit, il s'agira du débit mesuré in-situ, celui que le modèle tente de reproduire.
- La courbe en trait fin de couleur rouille correspond à la sortie simulée, c'est-à-dire à la somme contribution apporté par chaque branche, la somme des entrées convoluées par leur fonction de transfert respective.
- Les aires en bleu (du bleu foncé au bleu clair) représentent les contributions à la sortie totale des différentes branches, de manière cumulée. Chaque aire correspond au produit d'une entrée par sa fonction de transfert. Ces aires permettent d'identifier rapidement quelles entrées ont la plus grande contribution à la sortie.

Attention toutefois : MASH est avant tout un outil de traitement du signal et, comme on l'a vu précédemment, les problèmes étudiés ici sont « mal-posés ». Il est donc théoriquement possible de trouver différentes combinaisons d'entrées permettant de donner la même sortie et par conséquent l'affirmation consistant à dire que « la pluie contribue à X% du débit et la nappe à Y% » est erronée. La seule chose qu'il est possible d'affirmer avec certitude est que « la pluie permet d'expliquer X% du débit et la nappe Y% » et c'est tout.

Deux paramètres mesurant la qualité de l'ajustement sont affichés à titre indicatif dans le coin supérieur droit de cette partie : la somme des écarts quadratiques en haut et le coefficient de Nash-Sutcliffe en bas. Les deux paramètres sont inversement proportionnels à la qualité de l'ajustement. Le premier (compris entre zéro et l'infini) donne une mesure plus précise tandis que le second (compris entre moins l'infini et un) permet de comparer des problèmes ayant des nombres de données différents. Ils sont utiles pour les premières tentatives de calage du modèle en mode manuel, ainsi que pour vérifier le bon fonctionnement des algorithmes d'optimisation, en s'assurant que l'ajustement donné par l'algorithme et celui affiché dans l'interface graphique concordent.

A.3 Les fonctions de transfert et leurs paramètres

La zone droite de l'interface graphique est consacrée aux fonctions de transfert, représentées graphiquement dans la partie inférieure et dont les paramètres sont affichés dans la partie supérieure.

Les représentations graphiques des fonctions de transfert (**Figure A-4, a**) sont de la même couleur que les contributions à la sortie correspondantes sur le graphique des sorties, ce qui permet d'identifier rapidement quelle fonction de transfert engendre quelle contribution à la sortie. L'axe des ordonnées des graphiques correspond au gain des fonctions de transfert ; l'axe des abscisses au temps. À noter que les fonctions de transfert représentées sont les fonctions de transfert équivalentes, égales au produit de convolution des trois fonctions de transfert élémentaires de la branche considérée.

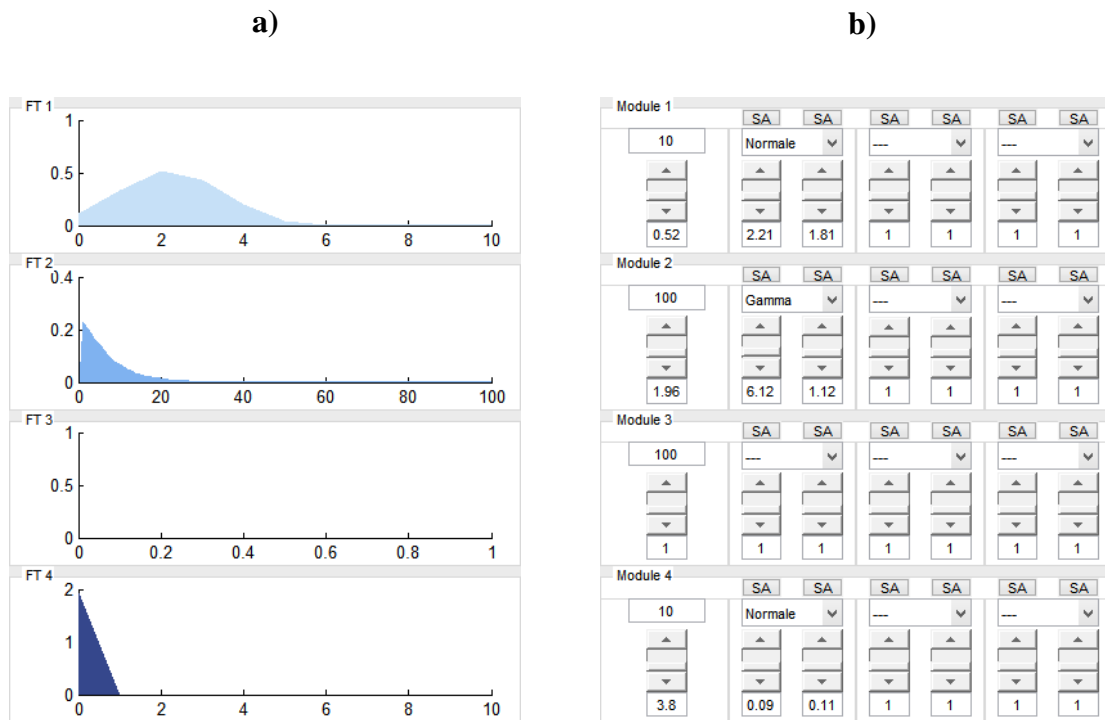


Figure A-4 : Détail de l'interface graphique utilisateur : (a) les fonctions de transfert ; (b) leurs paramètres

Au-dessus des représentations graphiques se trouvent les différents paramètres des fonctions de transfert. Chaque branche est composée de 8 zones de texte modifiable, de 7 sliders (qui modifient le même paramètre que la zone de texte modifiable qu'ils surmontent) et 3 menus déroulant. La partie la plus à gauche de la branche est consacrée au contrôle de la fonction de transfert équivalente : la zone de texte supérieure permet de régler sa longueur ; la zone de texte inférieure et le slider, son gain. Viennent ensuite trois blocs similaires, correspondant aux trois fonctions de transfert en série. Le menu déroulant permet de choisir la fonction de transfert parmi les cinq fonctions de transfert présentées précédemment (la chaîne de caractère « --- » indique qu'aucune fonction de transfert n'est sélectionnée). Les deux zones de texte ainsi que les sliders qui y sont associés permettent quant à eux de modifier la valeur des deux paramètres de la fonction de transfert sélectionnée. Les boutons « SA » permettent d'afficher la dérivée partielle de la fonction de transfert par rapport au paramètre correspondant. Pour finir, il est possible de masquer les zones de textes et les paramètres des sliders pour plus de lisibilité (Figure A-5).

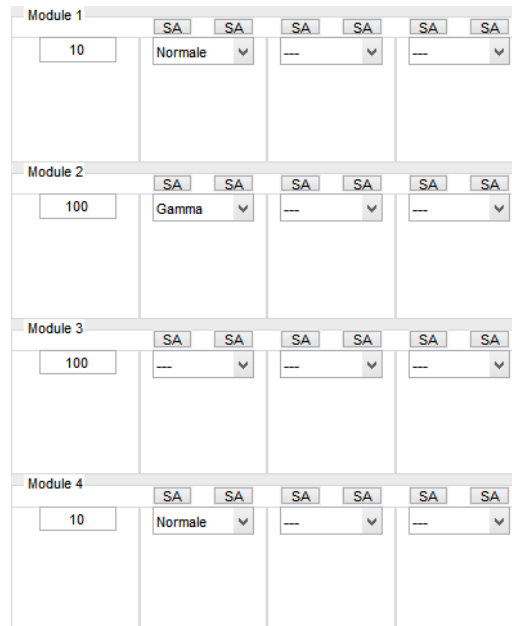


Figure A-5 : Détail de l'interface graphique utilisateur : les paramètres des fonctions de transfert en mode « Automatique »

Ce mode d'affichage, appelé « Automatique », est utilisé quand l'optimisation est effectuée à l'aide des algorithmes. Les seuls paramètres encore visibles sont ceux qui doivent être sélectionnés par l'utilisateur : la longueur de la fonction de transfert et le type de fonction de transfert utilisé.

A.4 Les menus

La fenêtre principale de MASH dispose de 5 menus : « Fichier », « Données », « Mode », « Inversion » et « Outils ». Les fonctions disponibles via ces différents menus sont les suivantes :

- Fichier :
 - Charger entrées/sortie : charge un jeu de données d'entrées et de sorties préalablement sauvegardé dans un fichier texte.
 - Charger fonctions de transfert : charge un jeu de paramètres de fonction de transfert préalablement sauvegardé dans un fichier texte.
 - Enregistrer entrées/sortie : enregistre un jeu de données d'entrées et de sorties dans un fichier texte.
 - Enregistrer fonctions de transfert : enregistre un jeu de paramètres de fonction de transfert dans un fichier texte.
 - Exporter données simulées : enregistre les contributions des différentes branches à la sortie simulée dans un fichier texte.

- Exporter sortie : enregistre la sortie (somme des différentes contributions) simulée dans un fichier texte.
- Export fonction de transfert : enregistre les fonctions de transfert dans un fichier texte.
- Données
 - Importer sortie : importe la chronique de sortie depuis un fichier Excel.
 - Importer entrée 1 : importe la chronique de l'entrée n°1 depuis un fichier Excel.
 - Importer entrée 2 : importe la chronique de l'entrée n°2 depuis un fichier Excel.
 - Importer entrée 3 : importe la chronique de l'entrée n°3 depuis un fichier Excel.
 - Importer entrée 4 : importe la chronique de l'entrée n°4 depuis un fichier Excel.
 - Importer ETP 1 : importe la chronique d'ETP n°1 depuis un fichier Excel.
 - Importer ETP 2 : importe la chronique d'ETP n°2 depuis un fichier Excel.
 - Importer ETP 3 : importe la chronique d'ETP n°3 depuis un fichier Excel.
 - Importer ETP 4 : importe la chronique d'ETP n°4 depuis un fichier Excel.
 - Importer toutes les entrées : importe simultanément les quatre chroniques d'entrée depuis un fichier Excel.
 - Importer toutes les ETP : importe simultanément les quatre chroniques d'ETP depuis un fichier Excel.
- Mode
 - Automatique : masque tous les paramètres des fonctions de transfert optimisés par les algorithmes PSO
 - Manuel : affiche tous les paramètres des fonctions de transfert optimisés par les algorithmes PSO
- Inversion :
 - Préparer PEST : créer le fichier de configuration nécessaire au bon fonctionnement de PEST
 - Lancer PEST : démarre l'optimisation des paramètres grâce au programme externe PEST. Les paramètres finaux des fonctions de transfert sont sauvegardés dans un fichier texte.
 - Paramètres PSO : ouvre une fenêtre contextuelle permettant de choisir le nombre de particules et le nombre itérations utilisés dans par les algorithmes d'optimisation PSO
 - Lancer CP-PSO : démarre l'optimisation des paramètres grâce au solveur interne utilisant l'algorithme « centered-progressive PSO ». Les paramètres finaux des fonctions de transfert (meilleure position de l'essaim) sont sauvegardés dans un fichier texte (valable pour les trois algorithmes PSO).
 - Lancer CC-PSO : démarre l'optimisation des paramètres grâce au solveur interne utilisant l'algorithme « centered-centered PSO ».

- Lancer CR-PSO : démarre l'optimisation des paramètres grâce au solveur interne utilisant l'algorithme « centered-regressive PSO »
- Charger paramètres des fonctions de transfert calculées. Charge les paramètres des fonctions de transfert optimisés par un des algorithmes depuis un fichier texte.
- Outils
 - Tracer résidus : ouvre une nouvelle fenêtre dans laquelle sont tracés trois nuages de points, chacun dans un graphique : les résidus (différence entre les données mesurées et les données simulées à un temps t) en fonction du temps, les résidus en fonction des données mesurées et enfin les résidus en fonctions des données simulées.
 - n PSO : lance n fois l'algorithme PSO à la suite, afin de réaliser des tests statistiques sur les simulations. Les paramètres finaux des fonctions de transfert de chaque simulation sont sauvegardés dans un fichier texte.

ANNEXE B : COURBES SUPPLÉMENTAIRES RELATIVES AU CAS RÉEL

B.1 Évolution de l'ajustement (R^2) pour les différents algorithmes PSO

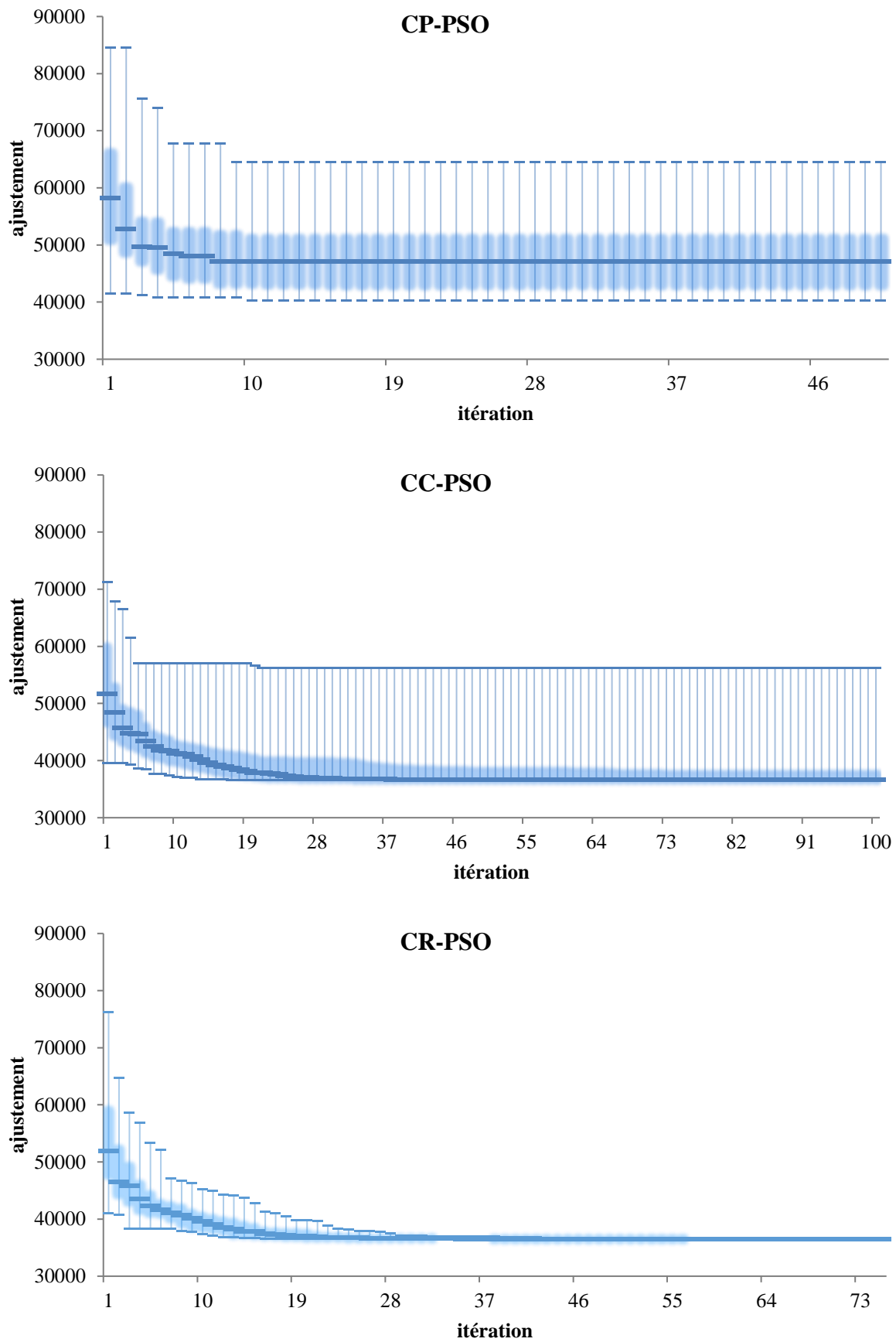


Figure B-1 : Modèle à 3 paramètres — 25 particules

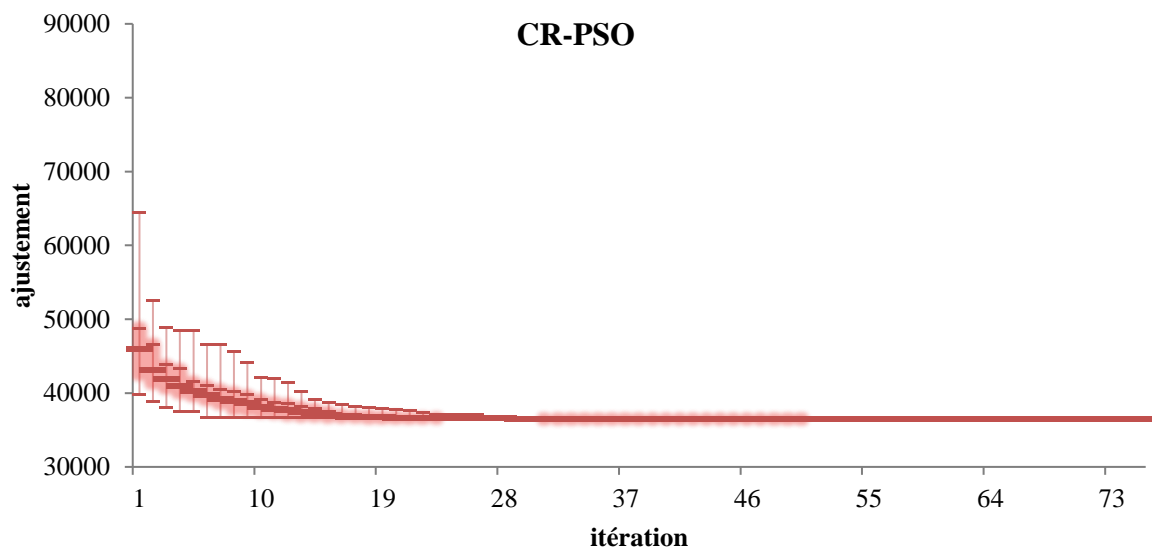
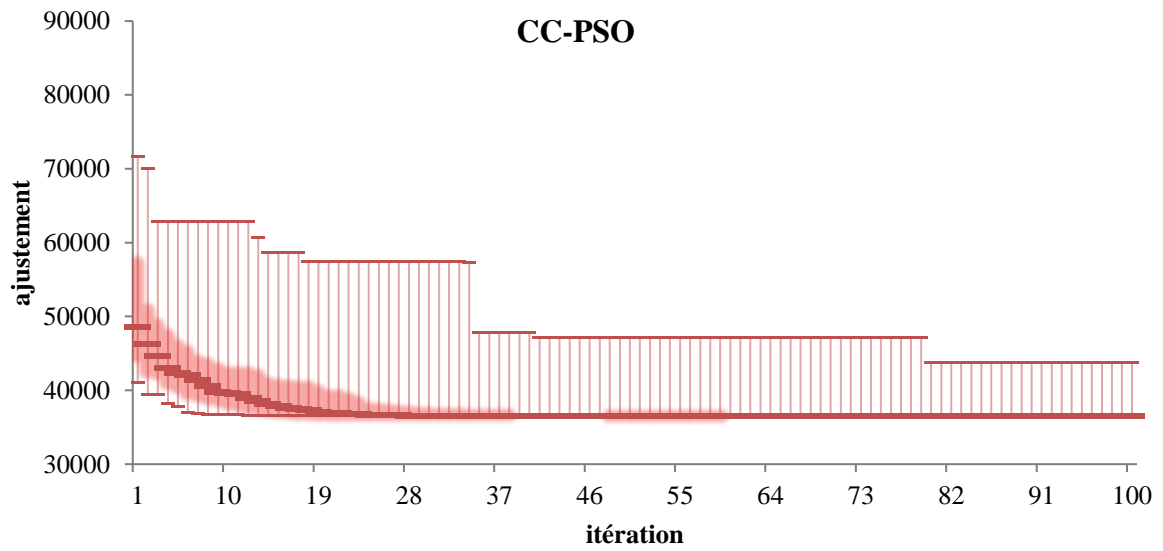
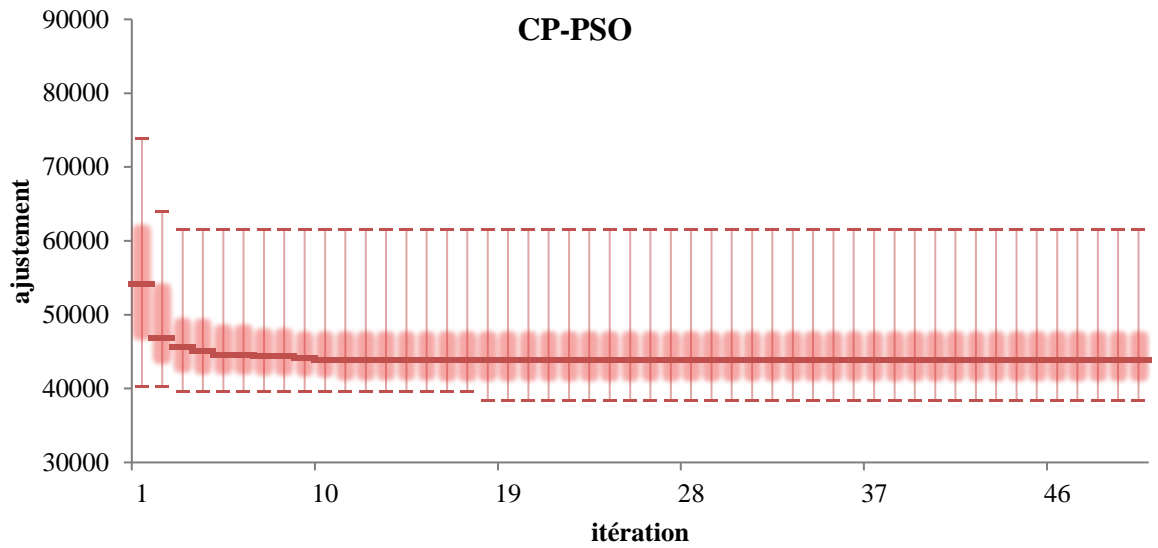


Figure B-2 : Modèle 3 paramètres — 50 particules

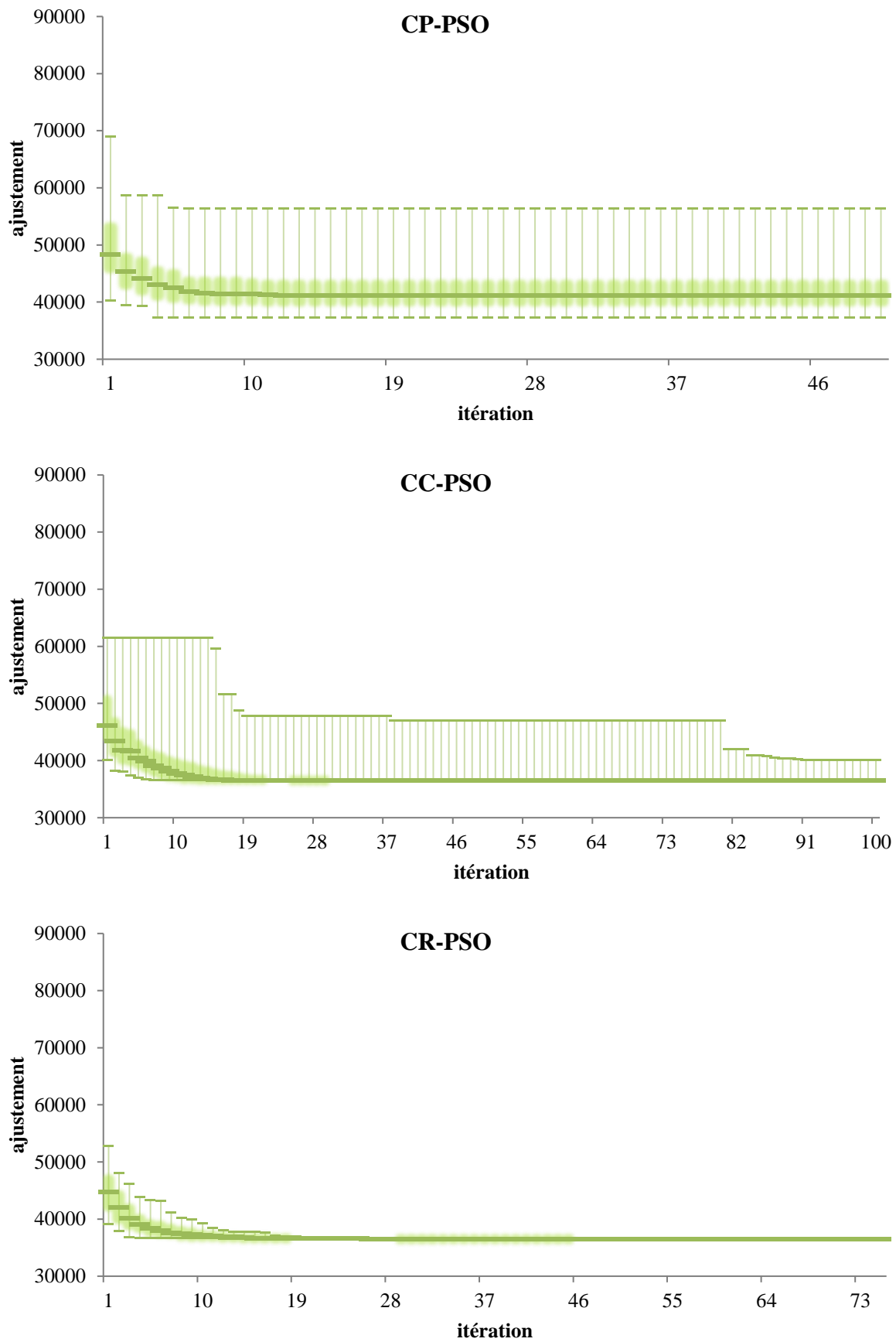


Figure B-3 : Modèle à 3 paramètres — 100 particules

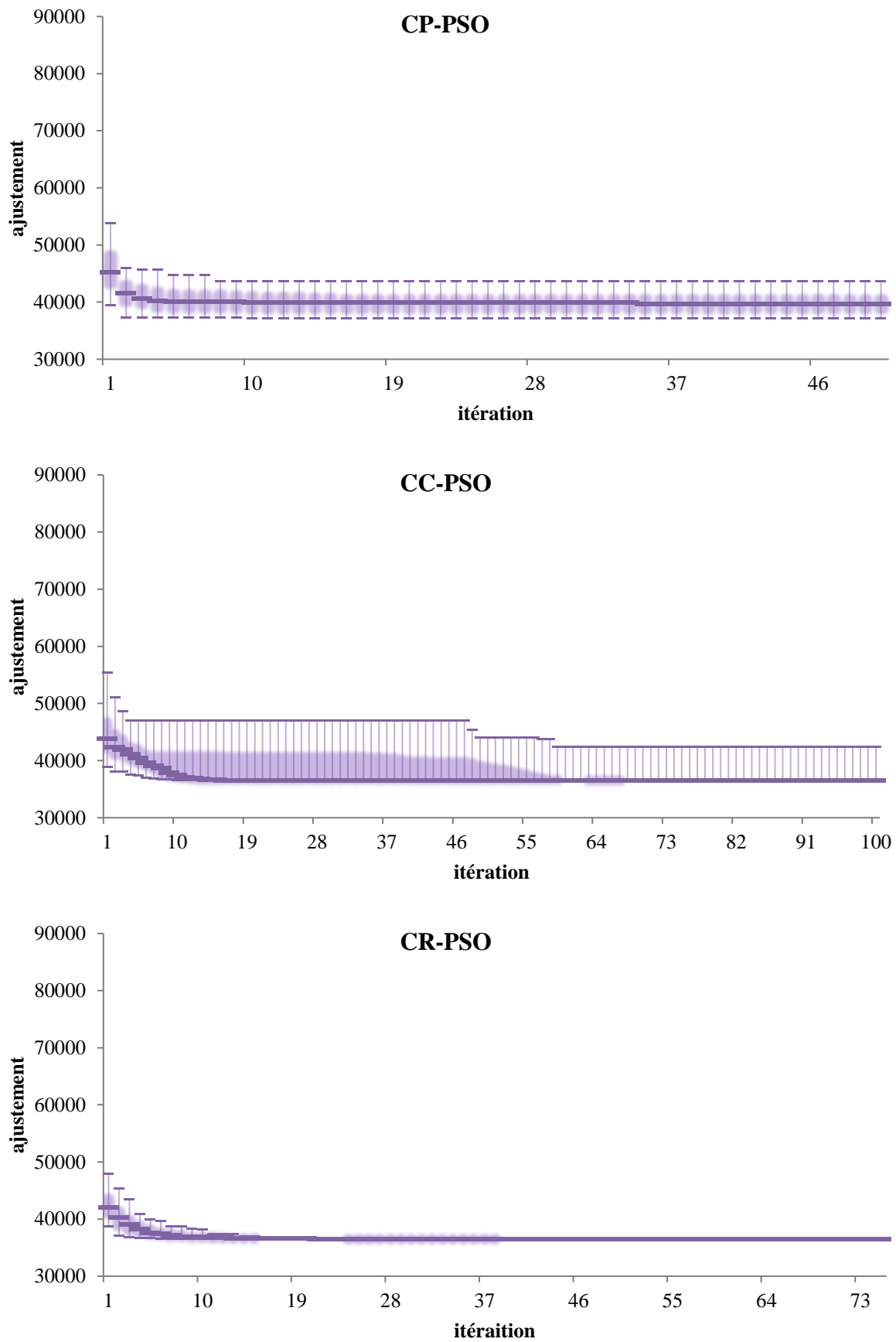


Figure B-4 : Modèle à 3 paramètres — 200 particules

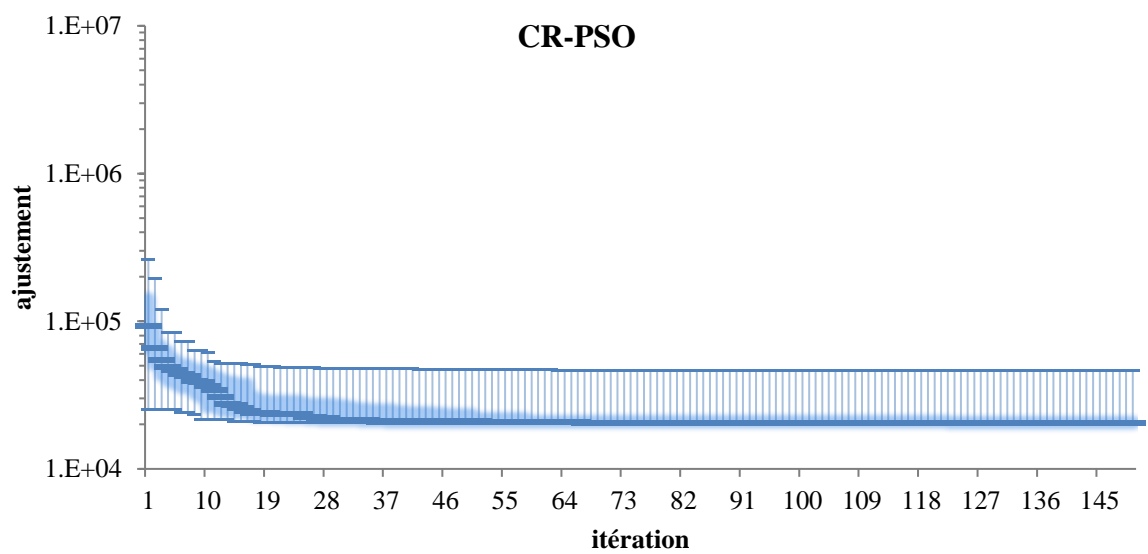
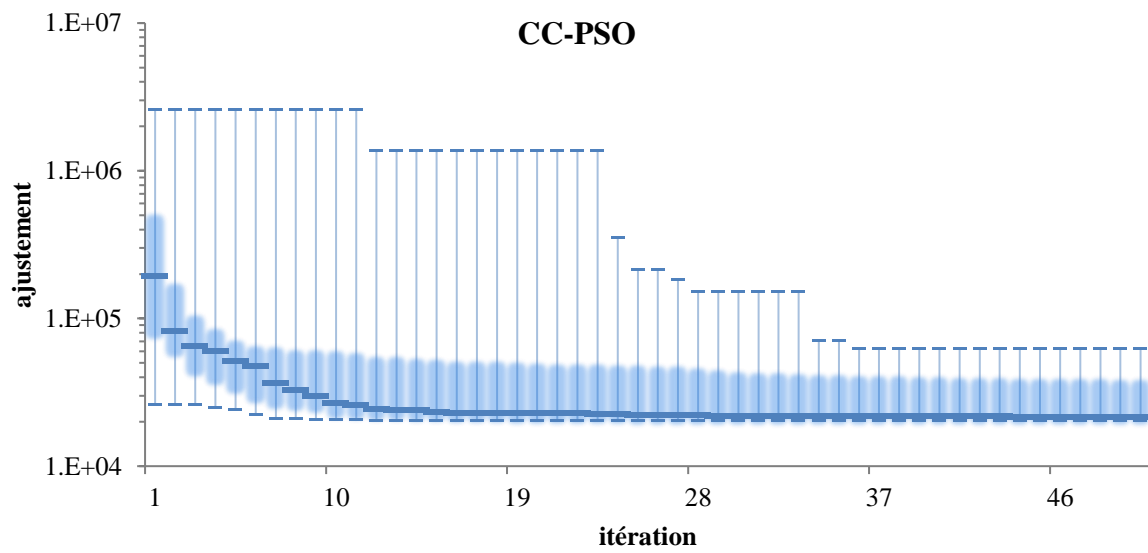
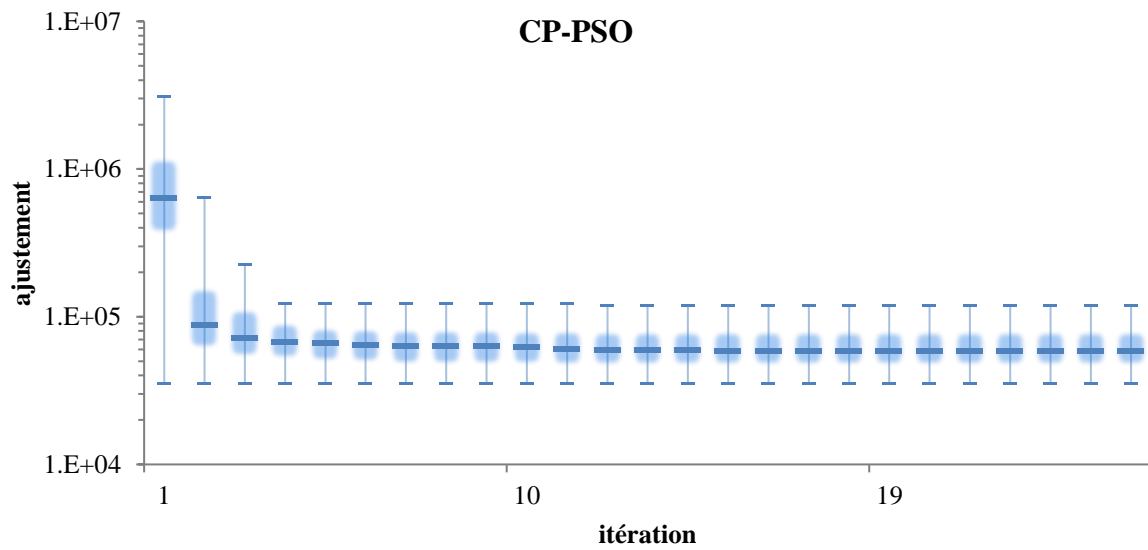


Figure B-5 : modèles à 6 paramètres — 25 particules

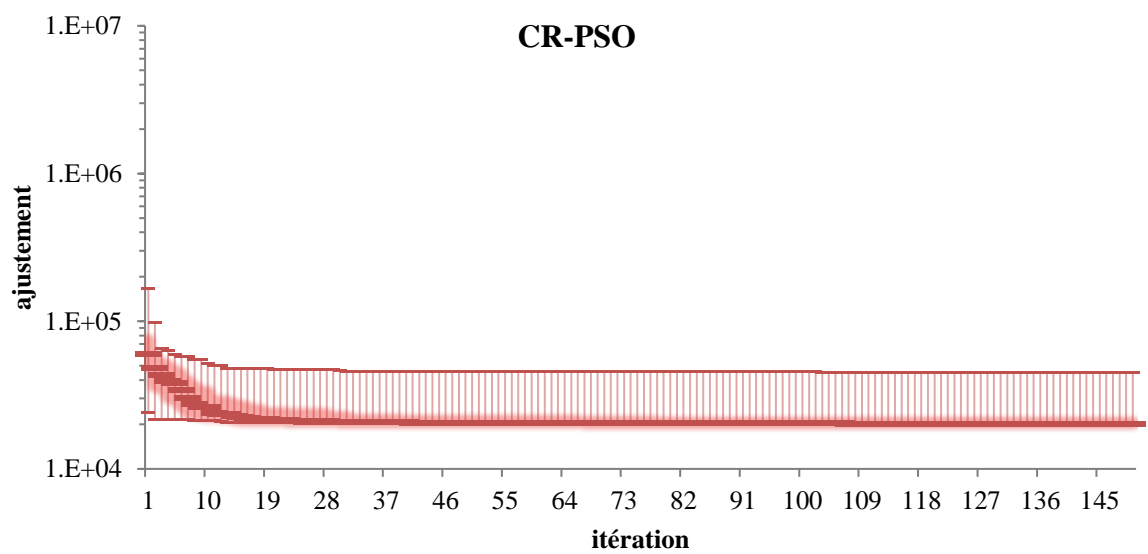
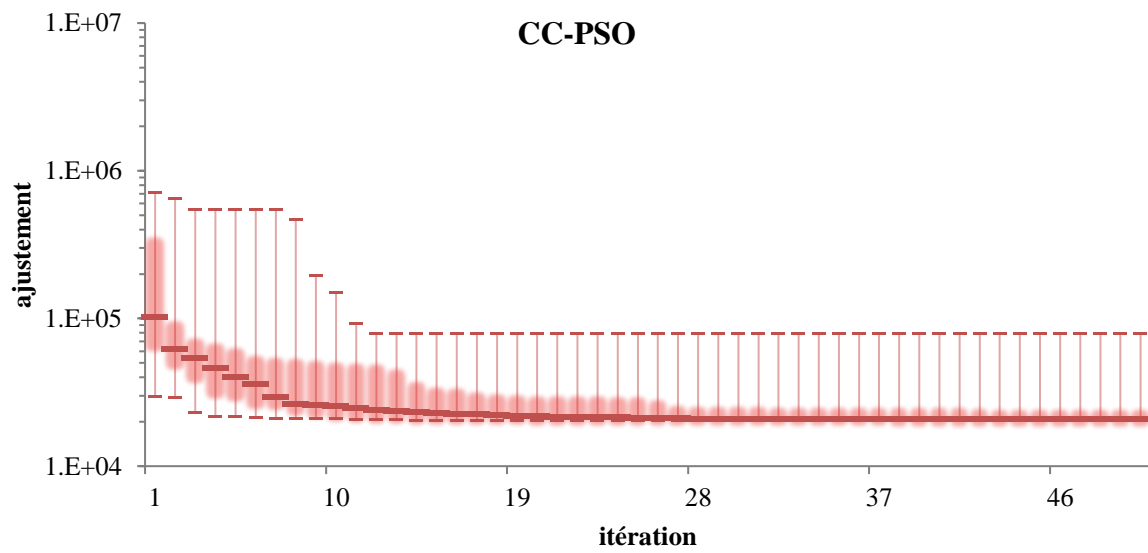
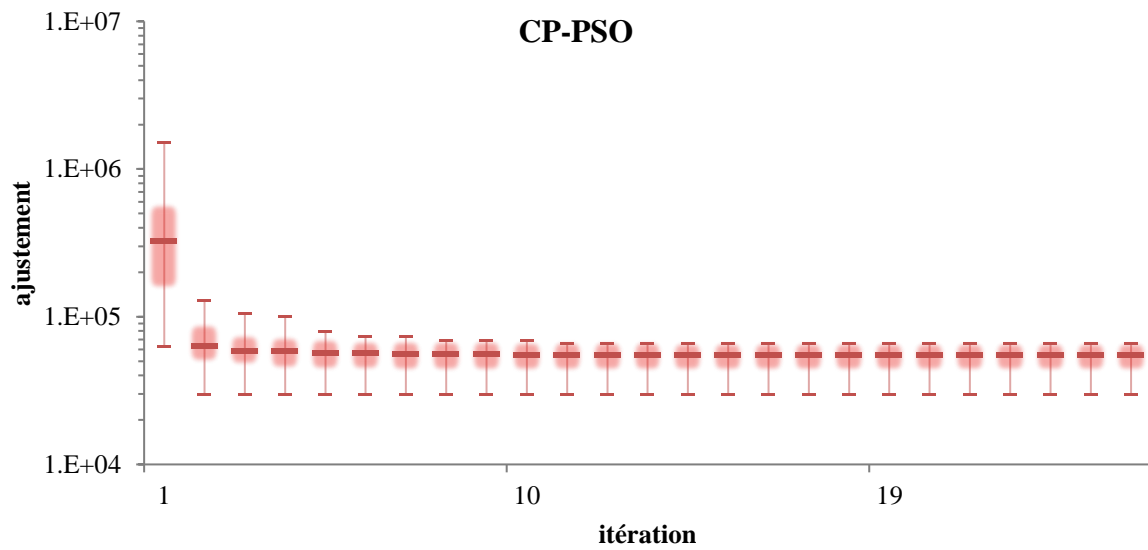


Figure B-6 : Modèle à 6 paramètres — 50 particules

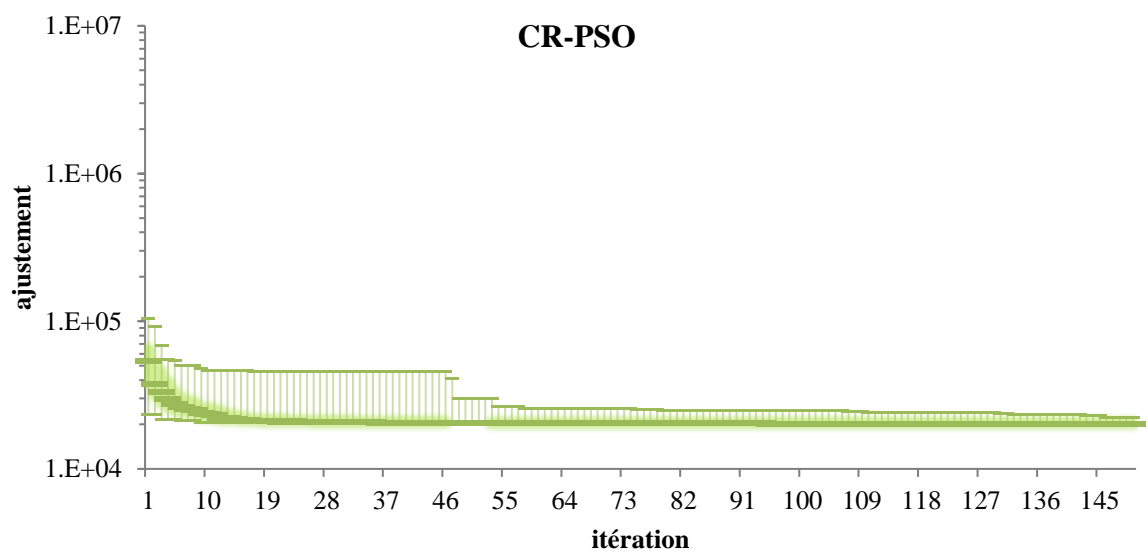
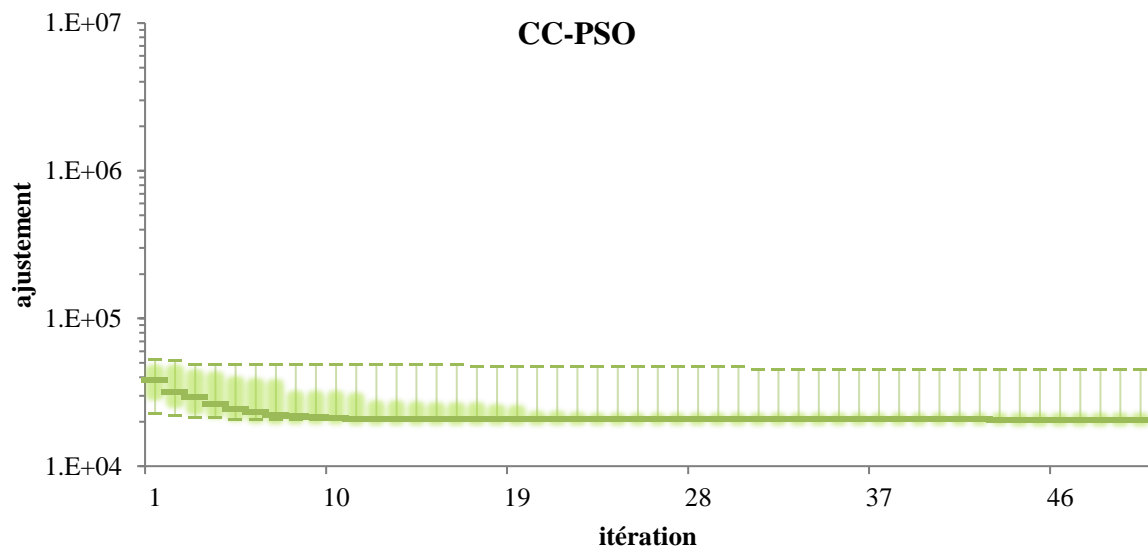
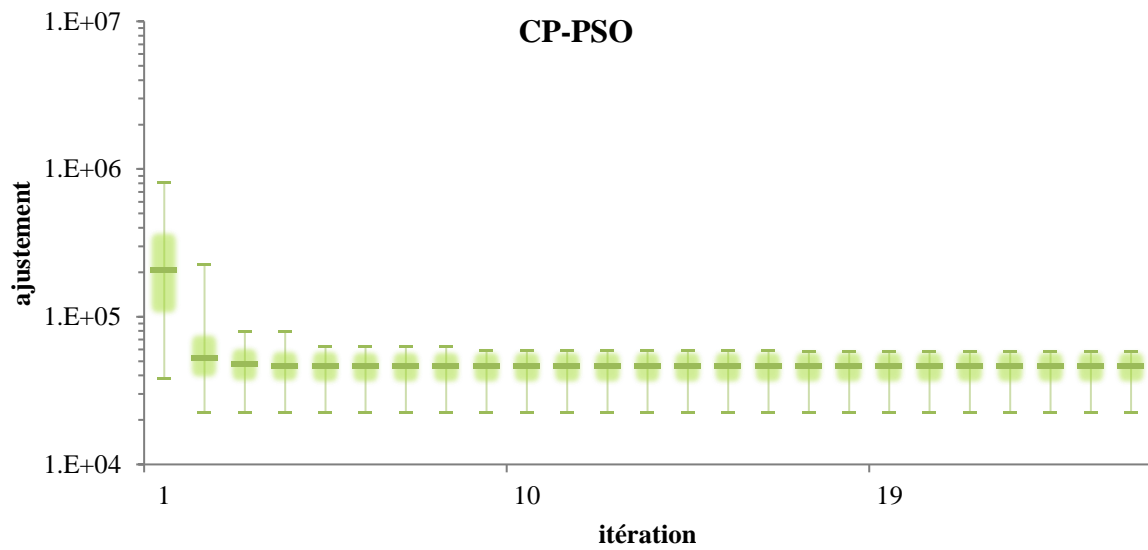


Figure B-7 : Modèle à 6 paramètres — 100 particules

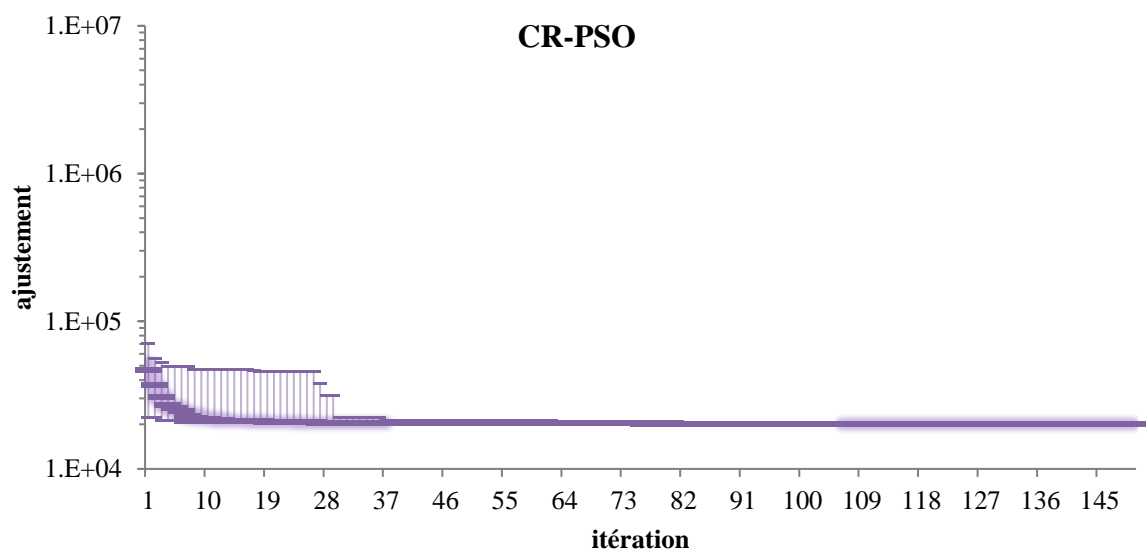
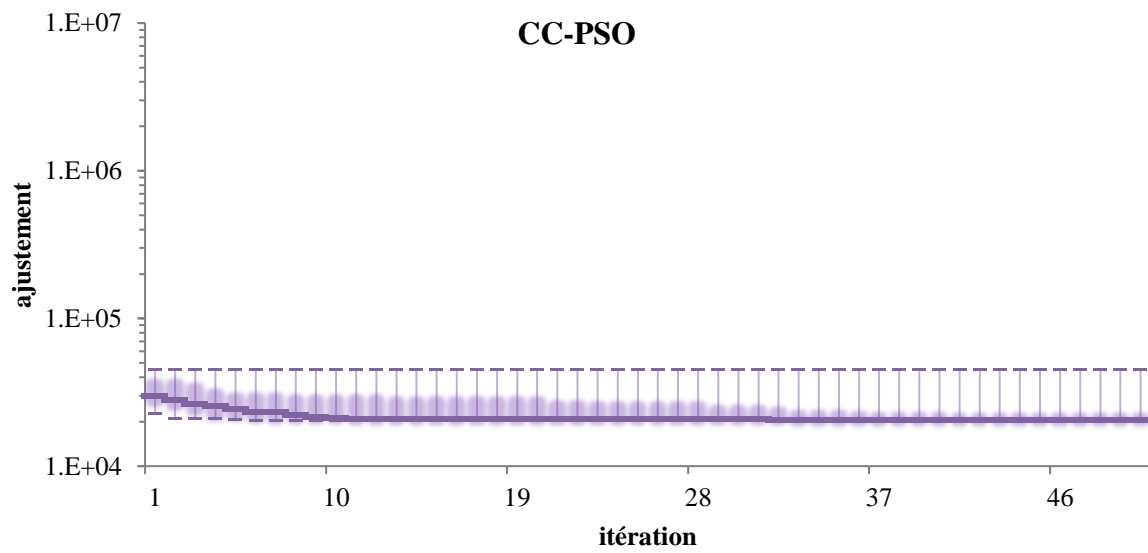
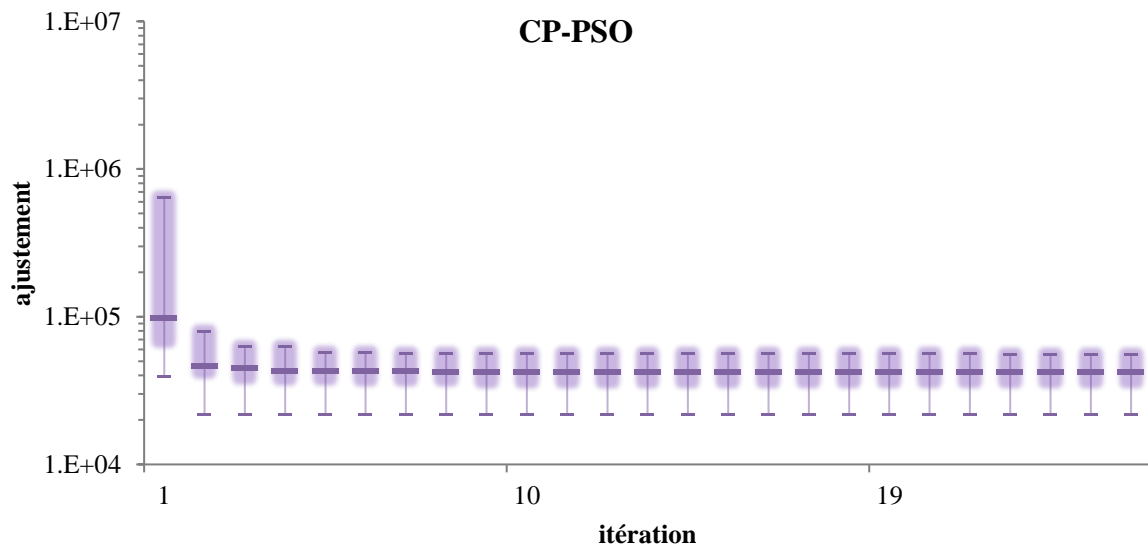


Figure B-8 : Modèle à 6 paramètres — 200 particules

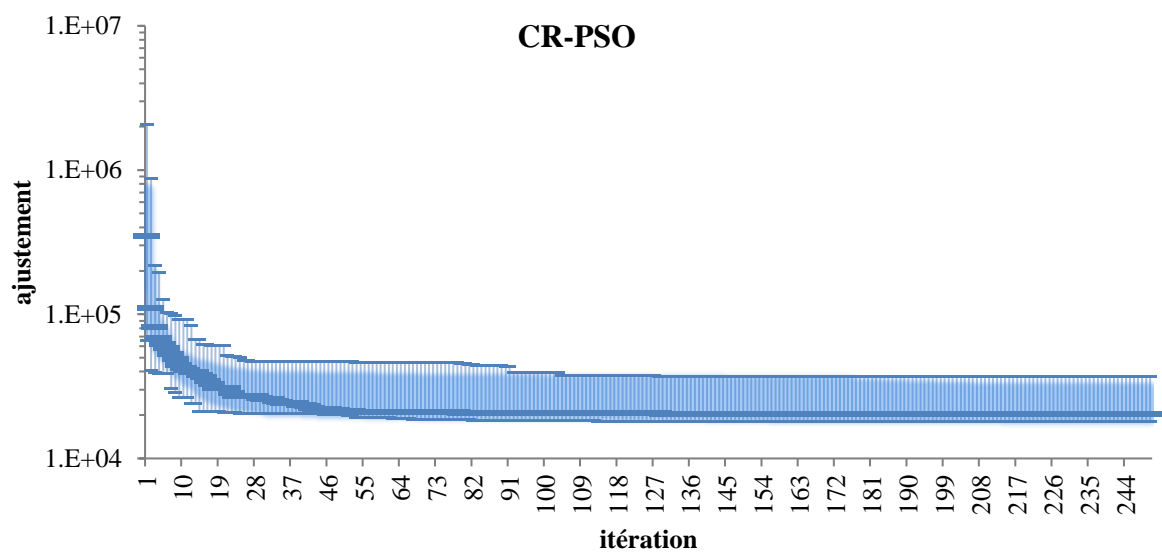
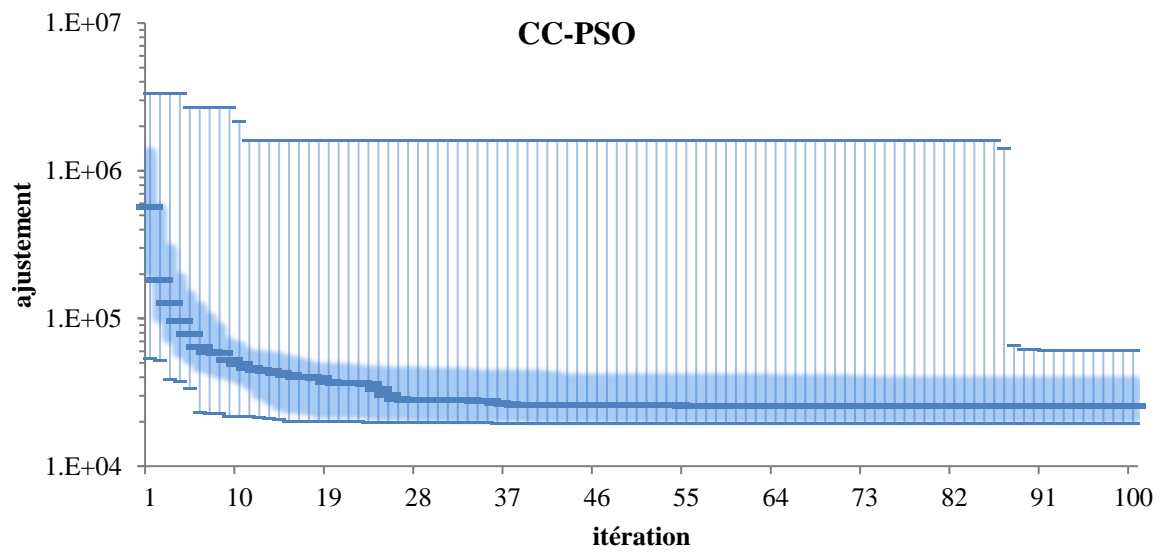
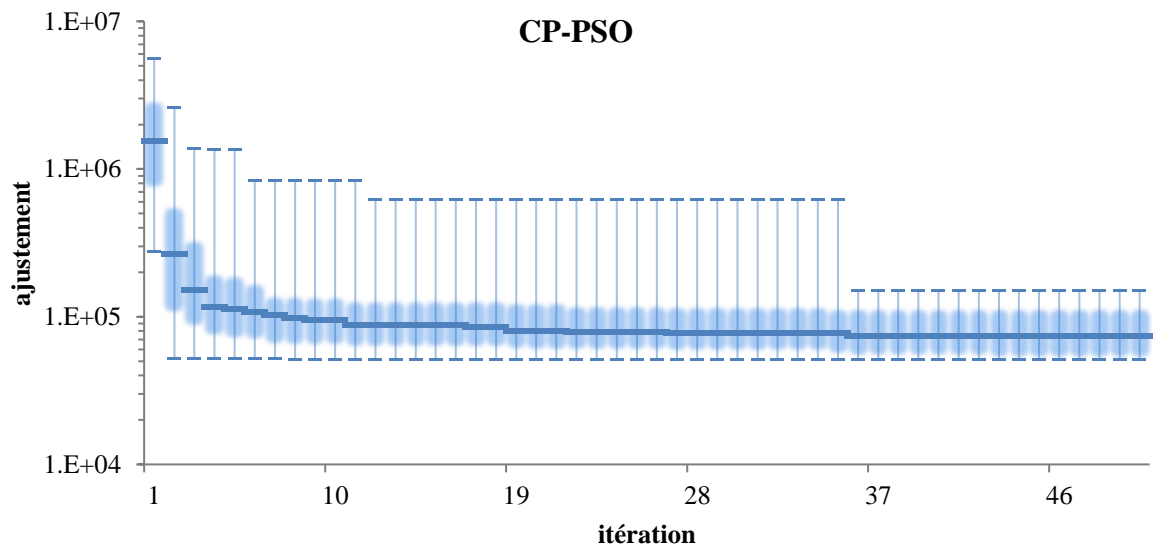


Figure B-9 : Modèle à 9 paramètres — 25 particules

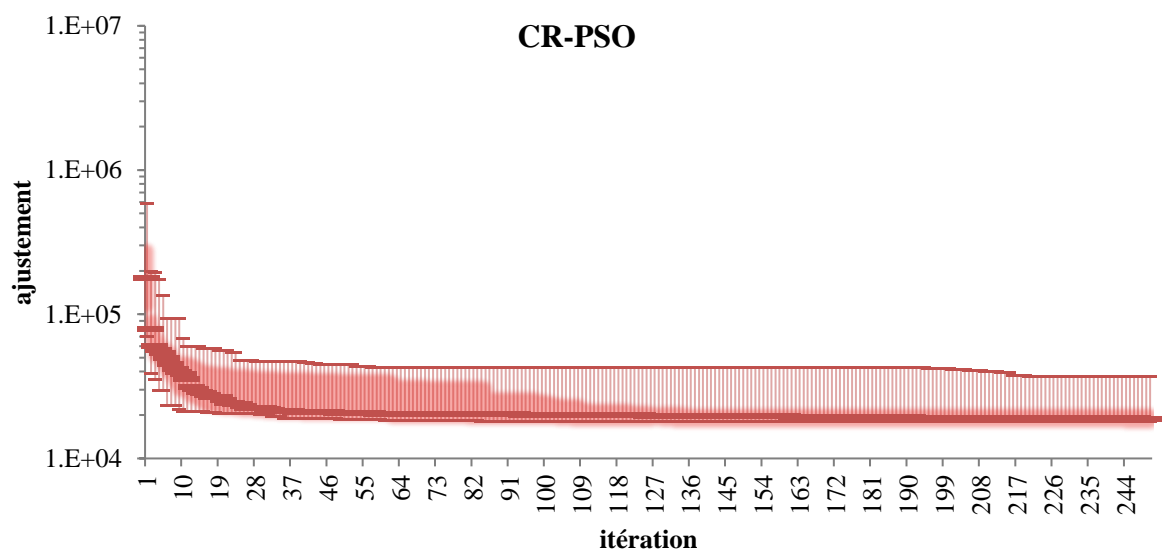
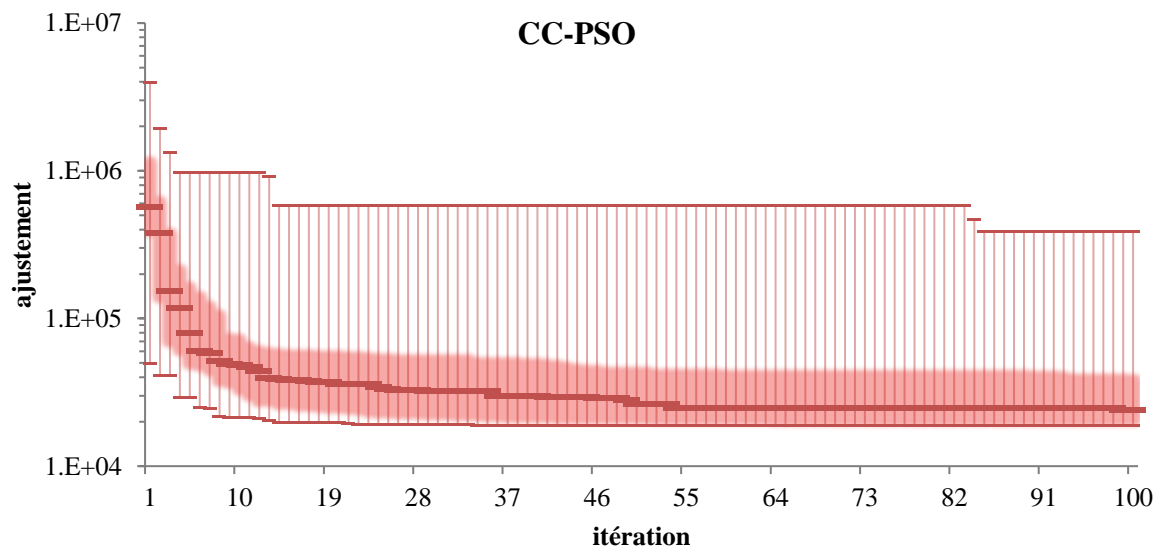
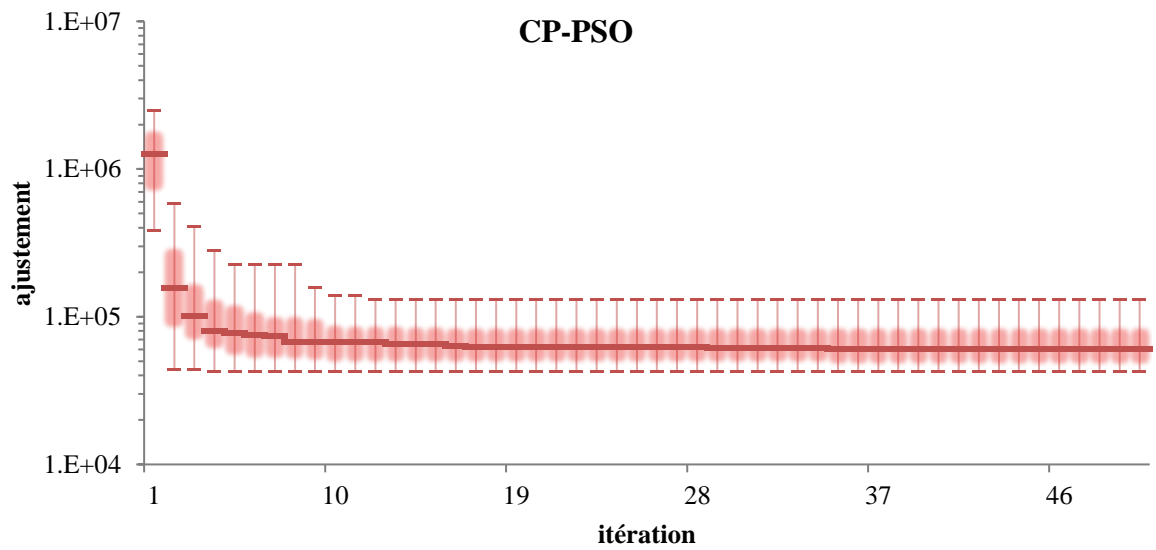


Figure B-10 : Modèle à 9 paramètres — 50 particules

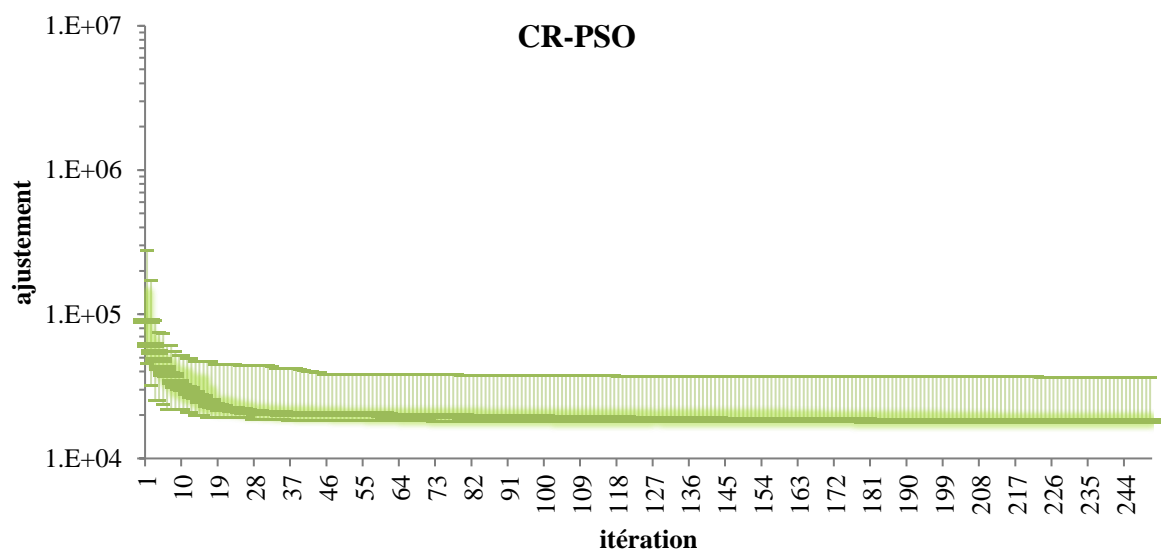
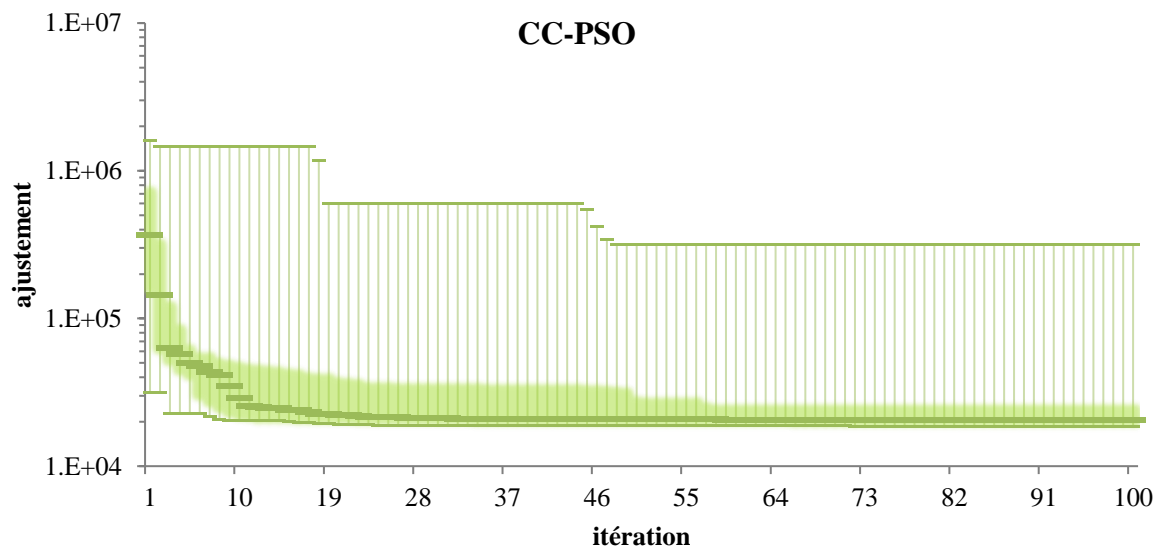
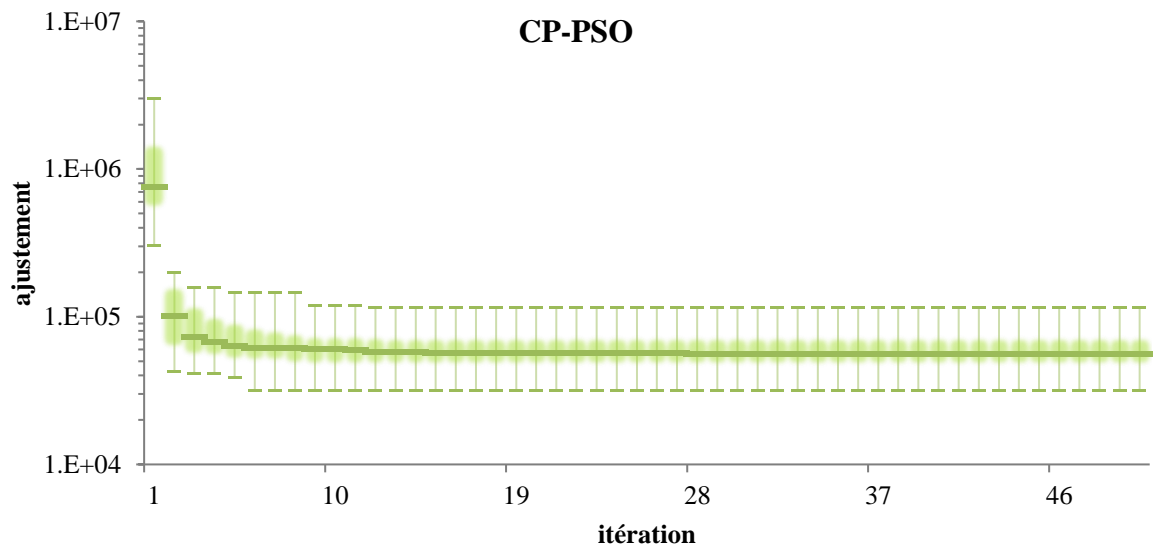


Figure B-11 : Modèle à 9 paramètres — 100 particules

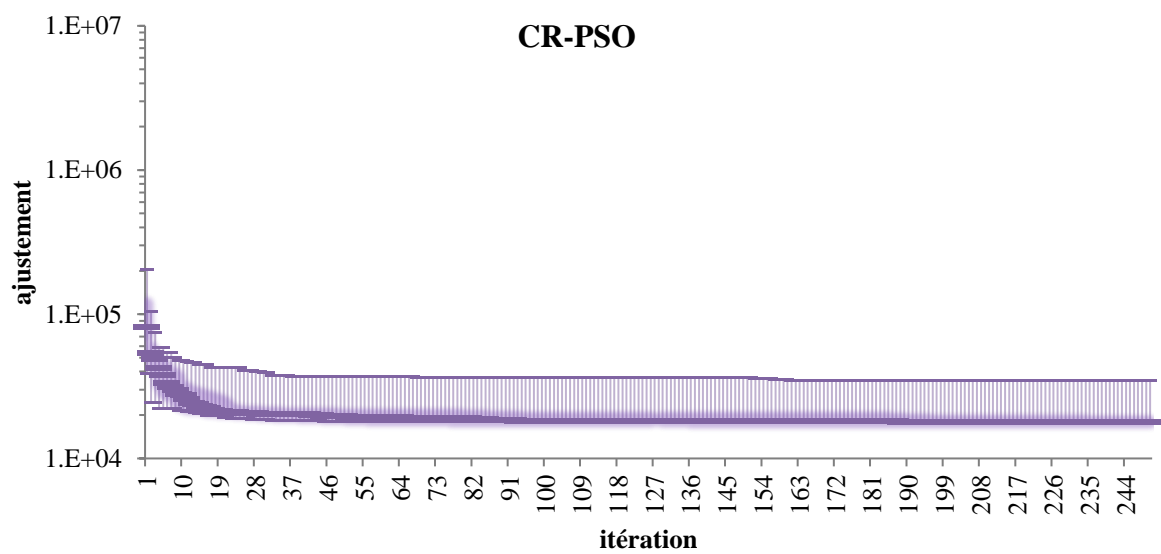
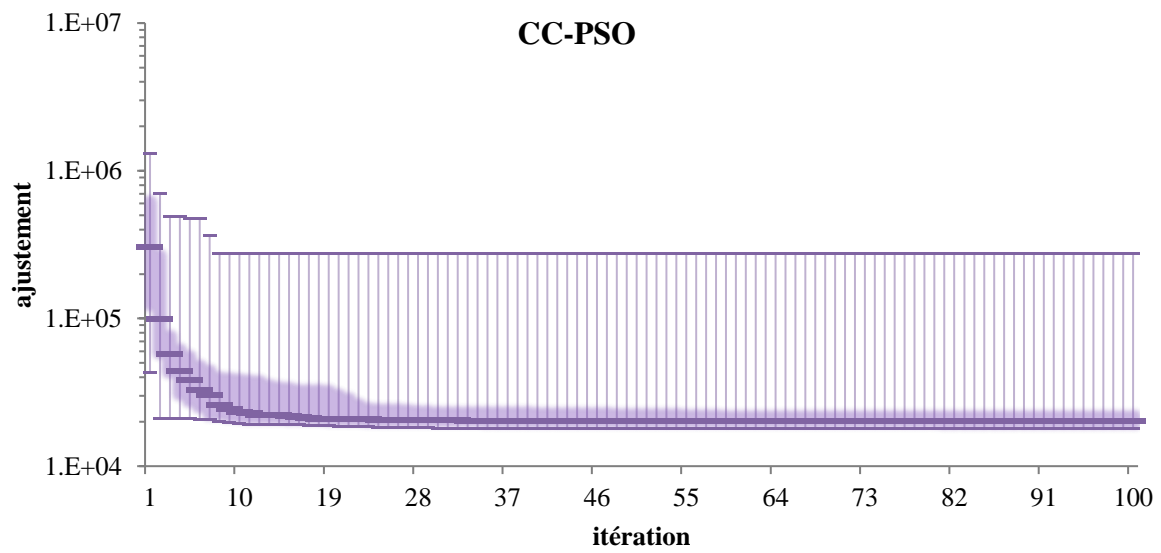
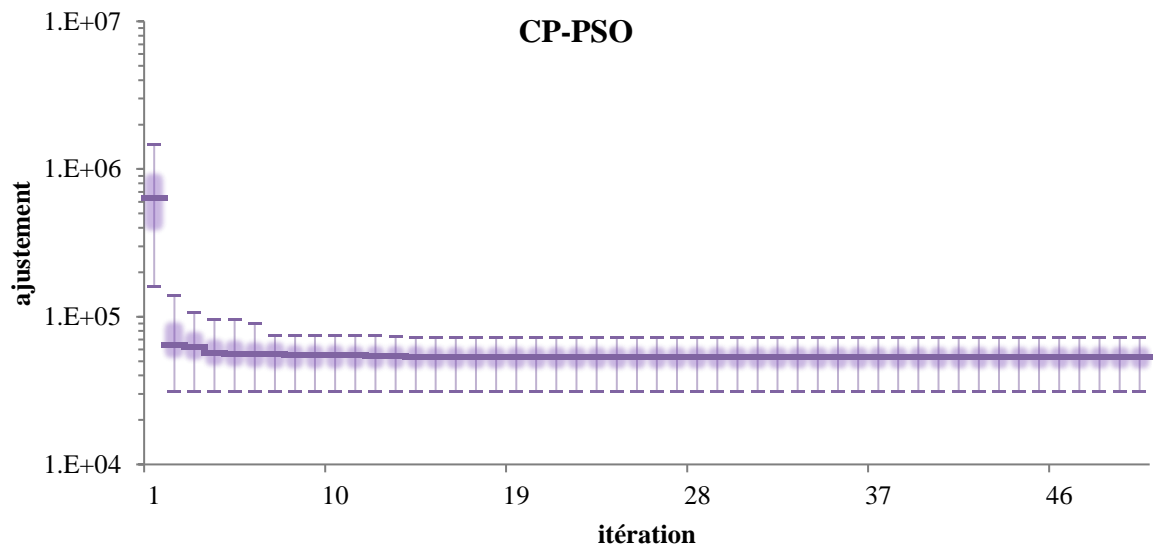


Figure B-12 : Modèle à 9 paramètres — 200 particules

B.2 Ajustements (Nash) pour les différentes architectures de MASH

Tableau B-1 : Ajustements (coefficients de Nash-Sutcliffe) des prédictions sur la période C, pour des modèles entraînés respectivement sur les périodes C, P1, P2, P3 et P4.

C : période complète (1993 – 2005) — P_i : période i — G : loi Gamma — N : loi normale — B : loi Béta — P : loi puissance — Fonctions de transfert en série avec les initiales juxtaposées (exemple : PN = loi puissance et loi normale en série) — Fonctions de transfert en parallèle avec les initiales séparées par un tiret (exemple : G-P = loi Gamma et loi puissance en parallèle).

C_prédiction									
C_calibration		P1_calibration		P2_calibration		P3_calibration		P4_calibration	
0.849	GP-PN-N	0.750	GG-PG-N	0.811	GP-PN-N	0.814	GBN-GBG-N	0.728	GG-PN-N
0.848	GPP-PNN-N	0.811	GPP-NBN	0.801	GPP-PPP	0.804	GP-GN-N	0.810	GP-PP
0.847	GPP-PNP-N	0.813	GPP-NBP	0.811	GPP-PPP-N	0.807	GP-GP-N	0.810	GP-PP-N
0.847	GB-PP-N	0.787	GB-PP-N	0.628	GB-PP-N	0.826	GB-GP-N	0.416	GB-PP-N
0.847	GB-PN-N	0.780	GB-PG-N	0.630	GB-PN-N	0.822	GB-GG-N	0.819	GB-PN-N
0.847	GPN-PNP-N	0.782	GPG-NBP	0.808	GPN-PPP-N	0.809	GN-GP-N	0.815	GPN-PPP-N
0.847	GPB-PNN-N	0.779	GN-PN-N	0.574	GPB-PPP	0.811	GBP-GBN-N	0.810	GN-PP
0.847	GPB-PNP-N	0.780	GN-PP-N	0.814	GPB-PPP-N	0.808	GBP-GBP-N	0.763	GN-PP-N
0.847	GN-PP-N	0.608	GBP-PPP-N	0.810	GN-PP-N	0.780	GBG-GBP-N	0.549	GBP-PNP
0.846	GB-PG-N	0.818	GB-NP	0.614	GB-PG-N	0.809	GB-BP	0.408	GB-PG-N
0.845	G-P-N	0.780	G-P-N	0.811	G-P-N	0.824	G-P-N	0.768	G-P-N
0.845	G-N-N	0.780	G-N-N	0.811	G-N-N	0.788	G-N-N	0.768	G-N-N
0.845	GP-PP-N	0.756	GG-PP-N	0.811	GP-PP-N	0.811	GBN-GBP-N	0.728	GG-PP-N
0.844	GP-PG-N	0.777	GG-NP	0.790	GP-PG-N	0.809	GBN-BGP	0.789	GG-PG-N
0.844	GPN-PNN-N	0.776	GPG-NBN	0.802	GPN-PPP	0.792	GN-GN-N	0.811	GPN-PPN-N
0.844	GPN-PNG-N	0.771	GPG-NBG	0.813	GPN-PPN-N	0.803	GN-GG-N	0.779	GPN-PPG-N
0.844	G-G-N	0.780	G-G-N	0.811	G-G-N	0.804	G-G-N	0.775	G-G-N
0.843	GN-PN-N	0.783	GBP-PPG-N	0.816	GN-PN-N	0.777	GBG-GBG-N	0.526	GBP-PNG
0.843	GN-PG-N	0.607	GBN-PPP-N	0.810	GN-PG-N	0.767	GBG-BGP	0.815	GBN-PNP
0.843	GPP-PNG-N	0.811	GPP-NBG	0.811	GPP-PPN-N	0.806	GP-GG-N	0.784	GP-PN-N
0.842	G-B-N	0.780	G-B-N	0.812	G-B-N	0.808	G-B-N	0.806	G-B-N
0.839	GPP-PNB-N	0.813	GPP-NBB	0.801	GPP-PPN	0.799	GP-GB-N	0.810	GP-PN
0.838	GB-PB-N	0.638	GB-NG	0.787	GB-PB-N	0.811	GB-BG	0.746	GB-PB-N
0.838	GPB-PNB-N	0.767	GN-PB-N	0.806	GPB-PPN	0.812	GBP-GBB-N	0.810	GN-PN
0.830	GP-PG	0.776	GG-NN	0.802	GP-PG	0.808	GBN-BGN	0.765	GG-PG
0.830	GP-PB-N	0.776	GG-NG	0.809	GP-PB-N	0.797	GBN-BGG	0.677	GG-PB-N
0.830	GP-PP	0.608	GG-PN-N	0.801	GP-PP	0.811	GBN-GBN-N	0.776	GG-PP
0.829	GPP-PGP	0.812	GPN-NBP	0.810	GPP-PPG-N	0.806	GP-BP	0.767	GP-PG-N
0.829	GPN-PGP	0.813	GPB-NBP	0.805	GPN-PPG-N	0.781	GN-BP	0.810	GNN-PPP-N
0.829	N-P-N	0.763	N-P-N	0.807	N-P-N	0.789	N-P-N	0.742	N-P-N

0.829	GP-PN	0.757	GG-PB-N	0.801	GP-PN	0.806	GBN-GBB-N	0.776	GG-PN
0.829	N-N-N	0.763	N-N-N	0.789	N-N-N	0.799	N-N-N	0.744	N-N-N
0.828	GPP-PGN	0.809	GPN-NBN	0.803	GPP-PPG	0.806	GP-BN	0.809	GP-PG
0.828	GB-PP	0.607	GB-PN-N	0.574	GB-PP	0.827	GB-GN-N	0.815	GB-PP
0.828	GB-PN	0.776	GB-PB-N	0.802	GB-PN	0.811	GB-GB-N	0.815	GB-PN
0.828	G-P	0.818	G-P	0.801	G-P	0.812	G-P	0.810	G-P
0.828	GB-PB	0.813	GB-NB	0.771	GB-PB	0.811	GB-BB	0.812	GB-PB
0.828	GPB-PGP	0.808	GN-NP	0.627	GPB-PPG-N	0.809	GBP-BGP	0.765	GN-PG-N
0.827	GPP-PGG	0.814	GPN-NBG	0.778	GPP-PPB-N	0.808	GP-BG	0.812	GP-PB-N
0.827	P-P-N	0.762	P-P-N	0.789	P-P-N	0.806	P-P-N	0.801	P-P-N
0.827	GPN-PGG	0.814	GPB-NBG	0.767	GPN-PPB-N	0.791	GN-BG	0.759	GNN-PPG-N
0.826	P-N-N	0.762	P-N-N	0.789	P-N-N	0.791	P-N-N	0.740	P-N-N
0.826	GN-PB	0.781	GBN-PPB-N	0.753	GN-PB	0.768	GBG-BGB	0.813	GBN-PNB
0.826	N-G-N	0.760	N-G-N	0.790	N-G-N	0.787	N-G-N	0.741	N-G-N
0.826	GN-PNB-N	0.769	GPG-NBB	0.804	GPN-PPN	0.786	GN-GB-N	0.807	GNN-PPB-N
0.826	GN-PN	0.783	GBP-PPB-N	0.801	GN-PN	0.763	GBG-GBB-N	0.811	GBP-PNB
0.825	GN-PP	0.601	GBP-PPN-N	0.801	GN-PP	0.769	GBG-GBN-N	0.808	GBP-PNN
0.825	G-N	0.809	G-N	0.801	G-N	0.797	G-N	0.810	G-N
0.825	G-B	0.811	G-B	0.803	G-B	0.793	G-B	0.810	G-B
0.825	GG-PP-N	0.607	GBG-PPP-N	0.629	GG-PP-N	0.814	GBB-GBP-N	0.547	GBG-PNP
0.825	G-G	0.809	G-G	0.803	G-G	0.793	G-G	0.809	G-G
0.825	GN-PG	0.610	GBN-PPN-N	0.804	GN-PG	0.769	GBG-BGN	0.545	GBN-PNN
0.825	GPG-PNP-N	0.780	GP-PP-N	0.630	GPG-PPP-N	0.804	GG-GP-N	0.766	GNG-PPP-N
0.824	P-G-N	0.760	P-G-N	0.785	P-G-N	0.793	P-G-N	0.790	P-G-N
0.824	GP-PB	0.777	GG-NB	0.803	GP-PB	0.808	GBN-BGB	0.773	GG-PB
0.823	GG-PN-N	0.589	GBG-PPG-N	0.784	GG-PN-N	0.814	GBB-GBG-N	0.528	GBG-PNG
0.823	N-B-N	0.762	N-B-N	0.806	N-B-N	0.781	N-B-N	0.779	N-B-N
0.823	GPB-PGB	0.811	GN-NB	0.782	GPB-PPB	0.809	GBP-BGB	0.810	GN-PB
0.823	GPN-PGN	0.815	GPB-NBN	0.804	GPN-PPG	0.798	GN-BN	0.810	GNN-PPN-N
0.823	GPP-PGB	0.813	GPN-NBB	0.794	GPP-PPB	0.806	GP-BB	0.811	GP-PB
0.822	GPN-PGB	0.816	GPB-NBB	0.780	GPN-PPB	0.795	GN-BB	0.785	GNN-PPB-N
0.819	GG-PG-N	0.608	GBB-PPP-N	0.778	GG-PG-N	0.810	GBB-BGP	0.544	GBB-PNP
0.817	GPG-PNG-N	0.786	GP-PG-N	0.768	GPG-PPN-N	0.773	GG-GG-N	0.410	GNG-PPG-N
0.814	B-P	0.745	B-P	0.729	B-P	0.808	B-P	0.739	B-P
0.814	B-B-N	0.734	B-B-N	0.774	B-B-N	0.778	B-B-N	0.791	B-B-N
0.813	P-B-N	0.749	P-B-N	0.751	P-B-N	0.792	P-B-N	0.798	P-B-N
0.812	GN-PB-N	0.592	GBN-PPG-N	0.774	GN-PB-N	0.772	GBG-BGG	0.811	GBN-PNG
0.807	B-B	0.803	B-B	0.750	B-B	0.755	B-B	0.753	B-B
0.806	N-N	0.796	N-N	0.777	N-N	0.797	N-N	0.787	N-N
0.806	N-P	0.799	N-P	0.777	N-P	0.787	N-P	0.787	N-P
0.804	N-G	0.795	N-G	0.776	N-G	0.773	N-G	0.784	N-G
0.797	GG-PB-N	0.591	GBB-PPG-N	0.743	GG-PB-N	0.809	GBB-BGG	0.522	GBB-PNG
0.797	P-P	0.792	P-P	0.776	P-P	0.786	P-P	0.782	P-P

0.796	P-N	0.791	P-N	0.776	P-N	0.771	P-N	0.783	P-N
0.796	GG-PG	0.608	GBB-PPN-N	0.763	GG-PG	0.810	GBB-BGN	0.549	GBB-PNN
0.795	N-B	0.790	N-B	0.777	N-B	0.769	N-B	0.783	N-B
0.794	P-G	0.780	P-G	0.776	P-G	0.773	P-G	0.780	P-G
0.793	B-N-N	0.728	B-N-N	0.753	B-N-N	0.776	B-N-N	0.663	B-N-N
0.793	B-P-N	0.728	B-P-N	0.753	B-P-N	0.818	B-P-N	0.657	B-P-N
0.792	GPG-PNB-N	0.776	GP-PB-N	0.578	GPG-PPN	0.773	GG-GB-N	0.705	GNG-PPB-N
0.792	GG-PP	0.600	GBG-PPN-N	0.769	GG-PP	0.810	GBB-GBN-N	0.544	GBG-PNN
0.792	GG-PN	0.717	GBG-PPB-N	0.768	GG-PN	0.813	GBB-GBB-N	0.767	GBG-PNB
0.790	B-G-N	0.726	B-G-N	0.750	B-G-N	0.814	B-G-N	0.663	B-G-N
0.789	GPG-PGP	0.820	GP-NP	0.782	GPG-PPG-N	0.776	GG-BP	0.428	GNB-PPP-N
0.786	PGB	0.775	PN	0.737	PGB	0.780	PGB	0.774	PGB
0.786	PGP	0.780	PNN	0.648	PGP	0.773	PGP	0.773	PGP
0.785	P-B	0.788	P-B	0.777	P-B	0.775	P-B	0.781	P-B
0.785	GPG-PGN	0.809	GP-NN	0.762	GPG-PPG	0.757	GG-BN	0.415	GNB-PPN-N
0.785	PG	0.719	PG	0.732	PG	0.775	PG	0.773	PG
0.784	PGN	0.781	PNG	0.732	PGN	0.777	PGN	0.770	PGN
0.783	GPG-PGG	0.814	GP-NG	0.756	GPG-PPB-N	0.747	GG-BG	0.405	GNB-PPG-N
0.782	PN	0.777	PNP	0.734	PN	0.763	PN	0.768	PN
0.781	GG-PB	0.779	GBB-PPB-N	0.740	GG-PB	0.810	GBB-BGB	0.810	GBB-PNB
0.779	GPG-PGB	0.812	GP-NB	0.770	GPG-PPB	0.758	GG-BB	0.815	GNB-PPB-N
0.766	B-N	0.764	B-N	0.729	B-N	0.742	B-N	0.764	B-N
0.760	PB	0.756	PB	0.713	PB	0.749	PB	0.745	PB
0.755	PP	0.749	PP	0.709	PP	0.747	PP	0.742	PP
0.749	B-G	0.742	B-G	0.749	B-G	0.744	B-G	0.794	B-G
0.742	PGG	0.780	PNB	0.686	PGG	0.629	PGG	0.728	PGG
0.734	P	0.729	P	0.691	P	0.724	P	0.721	P
0.715	B	0.710	B	0.673	B	0.700	B	0.699	B
0.702	GPG-PNN-N	0.785	GP-PN-N	0.769	GPG-PPP	0.772	GG-GN-N	0.433	GNG-PPN-N
0.685	GPB-PNG-N	0.741	GN-PG-N	0.632	GPB-PPN-N	0.812	GBP-GBG-N	0.814	GN-PN-N
0.684	G	0.680	G	0.648	G	0.672	G	0.666	G
0.678	N	0.671	N	0.638	N	0.667	N	0.656	N
0.589	GB-PG	0.818	GB-NN	0.554	GB-PG	0.810	GB-BN	0.526	GB-PG
0.586	GPB-PGN	0.808	GN-NN	0.557	GPB-PPG	0.807	GBP-BGN	0.809	GN-PG
0.573	GPB-PGG	0.811	GN-NG	0.790	GPB-PPB-N	0.808	GBP-BGG	0.754	GN-PB-N

Tableau B-2 : Ajustements (coefficients de Nash-Sutcliffe) des prédictions sur la période P1, pour des modèles entraînés respectivement sur les périodes C, P1, P2, P3 et P4.

C : période complète (1993 – 2005) — Pi : période i — G : loi Gamma — N : loi normale — B : loi Béta — P : loi puissance — Fonctions de transfert en série avec les initiales juxtaposées (exemple : PN = loi puissance et loi normale en série) — Fonctions de transfert en parallèle avec les initiales séparées par un tiret (exemple : G-P = loi Gamma et loi puissance en parallèle).

P1_prédiction									
C_calibration		P1_calibration		P2_calibration		P3_calibration		P4_calibration	
0.902	GB-PP-N	0.949	GB-PP-N	0.738	GB-PP-N	0.840	GB-GP-N	0.454	GB-PP-N
0.803	GPG-PNN-N	0.949	GP-PN-N	0.766	GPG-PPP	0.783	GG-GN-N	0.483	GNG-PPN-N
0.902	GB-PN-N	0.948	GB-PG-N	0.737	GB-PN-N	0.843	GB-GG-N	0.843	GB-PN-N
0.897	GN-PN-N	0.948	GBP-PPG-N	0.854	GN-PN-N	0.781	GBG-GBG-N	0.504	GBP-PNG
0.834	GN-PN	0.948	GBP-PPB-N	0.808	GN-PN	0.768	GBG-GBB-N	0.844	GBP-PNB
0.870	GPG-PNG-N	0.948	GP-PG-N	0.772	GPG-PPN-N	0.783	GG-GG-N	0.458	GNG-PPG-N
0.837	GN-PB	0.946	GBN-PPB-N	0.749	GN-PB	0.763	GBG-BGB	0.836	GBN-PNB
0.876	GPG-PNP-N	0.946	GP-PP-N	0.739	GPG-PPP-N	0.818	GG-GP-N	0.756	GNG-PPP-N
0.898	G-P-N	0.946	G-P-N	0.849	G-P-N	0.835	G-P-N	0.805	G-P-N
0.899	GPB-PNP-N	0.946	GN-PP-N	0.854	GPB-PPP-N	0.810	GBP-GBP-N	0.801	GN-PP-N
0.898	G-N-N	0.946	G-N-N	0.849	G-N-N	0.795	G-N-N	0.805	G-N-N
0.895	G-B-N	0.946	G-B-N	0.847	G-B-N	0.810	G-B-N	0.821	G-B-N
0.898	G-G-N	0.945	G-G-N	0.850	G-G-N	0.817	G-G-N	0.815	G-G-N
0.902	GPB-PNN-N	0.945	GN-PN-N	0.615	GPB-PPP	0.812	GBP-GBN-N	0.822	GN-PP
0.775	GG-PB	0.945	GBB-PPB-N	0.727	GG-PB	0.814	GBB-BGB	0.835	GBB-PNB
0.838	GB-PN	0.944	GB-PB-N	0.812	GB-PN	0.819	GB-GB-N	0.831	GB-PN
0.819	GPG-PNB-N	0.943	GP-PB-N	0.617	GPG-PPN	0.783	GG-GB-N	0.689	GNG-PPB-N
0.890	P-P-N	0.938	P-P-N	0.829	P-P-N	0.836	P-P-N	0.836	P-P-N
0.891	P-N-N	0.938	P-N-N	0.830	P-N-N	0.821	P-N-N	0.772	P-N-N
0.889	N-P-N	0.937	N-P-N	0.854	N-P-N	0.781	N-P-N	0.782	N-P-N
0.888	N-N-N	0.937	N-N-N	0.828	N-N-N	0.802	N-N-N	0.786	N-N-N
0.882	N-B-N	0.937	N-B-N	0.854	N-B-N	0.788	N-B-N	0.794	N-B-N
0.889	P-G-N	0.936	P-G-N	0.822	P-G-N	0.821	P-G-N	0.823	P-G-N
0.887	N-G-N	0.936	N-G-N	0.837	N-G-N	0.781	N-G-N	0.783	N-G-N
0.884	P-B-N	0.932	P-B-N	0.787	P-B-N	0.824	P-B-N	0.839	P-B-N
0.900	GPB-PNB-N	0.931	GN-PB-N	0.816	GPB-PPN	0.824	GBP-GBB-N	0.822	GN-PN
0.898	GP-PP-N	0.929	GG-PP-N	0.849	GP-PP-N	0.814	GBN-GBP-N	0.718	GG-PP-N
0.840	GP-PN	0.927	GG-PB-N	0.809	GP-PN	0.806	GBN-GBB-N	0.771	GG-PN
0.902	GP-PN-N	0.925	GG-PG-N	0.849	GP-PN-N	0.825	GBN-GBG-N	0.719	GG-PN-N
0.846	B-B-N	0.924	B-B-N	0.777	B-B-N	0.808	B-B-N	0.800	B-B-N
0.786	GPB-PNG-N	0.919	GN-PG-N	0.741	GPB-PPN-N	0.821	GBP-GBG-N	0.830	GN-PN-N
0.870	B-N-N	0.918	B-N-N	0.812	B-N-N	0.812	B-N-N	0.699	B-N-N
0.870	B-P-N	0.918	B-P-N	0.812	B-P-N	0.835	B-P-N	0.718	B-P-N
0.869	B-G-N	0.917	B-G-N	0.809	B-G-N	0.831	B-G-N	0.699	B-G-N
0.789	GG-PN	0.916	GBG-PPB-N	0.765	GG-PN	0.817	GBB-GBB-N	0.782	GBG-PNB
0.828	GG-PG	0.876	GBB-PPN-N	0.760	GG-PG	0.815	GBB-BGN	0.529	GBB-PNN

0.841	GP-PP	0.876	GG-PN-N	0.809	GP-PP	0.813	GBN-GBN-N	0.770	GG-PP
0.876	GG-PP-N	0.876	GBG-PPP-N	0.740	GG-PP-N	0.825	GBB-GBP-N	0.525	GBG-PNP
0.873	GG-PG-N	0.876	GBB-PPP-N	0.806	GG-PG-N	0.816	GBB-BGP	0.522	GBB-PNP
0.898	GN-PG-N	0.876	GBN-PPP-N	0.849	GN-PG-N	0.764	GBG-BGP	0.833	GBN-PNP
0.837	GB-PP	0.876	GB-PN-N	0.615	GB-PP	0.841	GB-GN-N	0.831	GB-PP
0.898	GN-PP-N	0.876	GBP-PPP-N	0.849	GN-PP-N	0.788	GBG-GBP-N	0.528	GBP-PNP
0.837	GN-PG	0.876	GBN-PPN-N	0.813	GN-PG	0.766	GBG-BGN	0.523	GBN-PNN
0.835	GN-PP	0.876	GBP-PPN-N	0.809	GN-PP	0.765	GBG-GBN-N	0.824	GBP-PNN
0.789	GG-PP	0.876	GBG-PPN-N	0.765	GG-PP	0.815	GBB-GBN-N	0.521	GBG-PNN
0.850	GG-PB-N	0.864	GBB-PPG-N	0.758	GG-PB-N	0.814	GBB-BGG	0.500	GBB-PNG
0.868	GN-PB-N	0.863	GBN-PPG-N	0.799	GN-PB-N	0.768	GBG-BGG	0.826	GBN-PNG
0.872	GG-PN-N	0.863	GBG-PPG-N	0.811	GG-PN-N	0.822	GBB-GBG-N	0.506	GBG-PNG
0.836	GPP-PGB	0.853	GPN-NBB	0.798	GPP-PPB	0.815	GP-BB	0.826	GP-PB
0.838	GPN-PGN	0.853	GPB-NBN	0.814	GPN-PPG	0.798	GN-BN	0.846	GNN-PPN-N
0.899	GPP-PNB-N	0.853	GPP-NBB	0.809	GPP-PPN	0.801	GP-GB-N	0.822	GP-PN
0.786	GPG-PGB	0.853	GP-NB	0.766	GPG-PPB	0.751	GG-BB	0.837	GNB-PPB-N
0.903	GPP-PNP-N	0.853	GPP-NBP	0.849	GPP-PPP-N	0.815	GP-GP-N	0.825	GP-PP-N
0.844	GPN-PGB	0.853	GPB-NBB	0.777	GPN-PPB	0.795	GN-BB	0.793	GNN-PPB-N
0.838	GB-PB	0.853	GB-NB	0.781	GB-PB	0.816	GB-BB	0.841	GB-PB
0.898	GPP-PNG-N	0.853	GPP-NBG	0.849	GPP-PPN-N	0.818	GP-GG-N	0.788	GP-PN-N
0.843	GPP-PGP	0.853	GPN-NBP	0.849	GPP-PPG-N	0.807	GP-BP	0.807	GP-PG-N
0.902	GPP-PNN-N	0.853	GPP-NBN	0.809	GPP-PPP	0.817	GP-GN-N	0.822	GP-PP
0.844	GPP-PGG	0.853	GPN-NBG	0.776	GPP-PPB-N	0.816	GP-BG	0.845	GP-PB-N
0.843	GPN-PGG	0.852	GPB-NBG	0.792	GPN-PPB-N	0.790	GN-BG	0.800	GNN-PPG-N
0.842	GPN-PGP	0.852	GPB-NBP	0.844	GPN-PPG-N	0.780	GN-BP	0.836	GNN-PPP-N
0.785	GPG-PGG	0.851	GP-NG	0.768	GPG-PPB-N	0.735	GG-BG	0.433	GNB-PPG-N
0.840	GPP-PGN	0.850	GPN-NBN	0.813	GPP-PPG	0.813	GP-BN	0.823	GP-PG
0.787	GPG-PGP	0.849	GP-NP	0.812	GPG-PPG-N	0.768	GG-BP	0.474	GNB-PPP-N
0.836	G-N	0.848	G-N	0.809	G-N	0.798	G-N	0.822	G-N
0.837	G-G	0.848	G-G	0.813	G-G	0.793	G-G	0.823	G-G
0.783	GPG-PGN	0.848	GP-NN	0.760	GPG-PPG	0.745	GG-BN	0.457	GNB-PPN-N
0.562	GPB-PGG	0.848	GN-NG	0.822	GPB-PPB-N	0.813	GBP-BGG	0.759	GN-PB-N
0.839	GPB-PGP	0.848	GN-NP	0.720	GPB-PPG-N	0.815	GBP-BGP	0.806	GN-PG-N
0.837	G-B	0.848	G-B	0.813	G-B	0.794	G-B	0.826	G-B
0.579	GPB-PGN	0.848	GN-NN	0.597	GPB-PPG	0.813	GBP-BGN	0.823	GN-PG
0.843	GPB-PGB	0.848	GN-NB	0.793	GPB-PPB	0.812	GBP-BGB	0.827	GN-PB
0.836	G-P	0.845	G-P	0.809	G-P	0.814	G-P	0.822	G-P
0.589	GB-PG	0.843	GB-NN	0.594	GB-PG	0.813	GB-BN	0.503	GB-PG
0.834	PGN	0.841	PNG	0.767	PGN	0.821	PGN	0.828	PGN
0.836	PGP	0.840	PNN	0.682	PGP	0.814	PGP	0.830	PGP
0.779	PGG	0.840	PNB	0.710	PGG	0.652	PGG	0.769	PGG
0.832	PN	0.839	PNP	0.773	PN	0.803	PN	0.826	PN
0.835	PGB	0.837	PN	0.772	PGB	0.820	PGB	0.831	PGB
0.819	N-N	0.829	N-N	0.783	N-N	0.801	N-N	0.802	N-N

0.819	N-G	0.829	N-G	0.785	N-G	0.784	N-G	0.801	N-G
0.902	GB-PG-N	0.827	GB-NP	0.721	GB-PG-N	0.815	GB-BP	0.455	GB-PG-N
0.819	N-P	0.825	N-P	0.783	N-P	0.790	N-P	0.802	N-P
0.819	B-B	0.823	B-B	0.758	B-B	0.755	B-B	0.776	B-B
0.805	P-G	0.814	P-G	0.787	P-G	0.778	P-G	0.797	P-G
0.810	PB	0.814	PB	0.750	PB	0.790	PB	0.802	PB
0.806	N-B	0.814	N-B	0.785	N-B	0.770	N-B	0.797	N-B
0.806	P-P	0.813	P-P	0.785	P-P	0.786	P-P	0.796	P-P
0.805	P-B	0.812	P-B	0.787	P-B	0.784	P-B	0.801	P-B
0.806	P-N	0.811	P-N	0.786	P-N	0.781	P-N	0.798	P-N
0.806	PP	0.809	PP	0.748	PP	0.790	PP	0.799	PP
0.902	GPN-PNG-N	0.793	GPG-NBG	0.853	GPN-PPN-N	0.814	GN-GG-N	0.799	GNP-PPG-N
0.900	GPN-PNP-N	0.793	GPG-NBP	0.849	GPN-PPP-N	0.806	GN-GP-N	0.834	GNP-PPP-N
0.863	GPN-PNB-N	0.792	GPG-NBB	0.811	GPN-PPN	0.789	GN-GB-N	0.842	GNP-PPB-N
0.898	GP-PG-N	0.790	GG-NP	0.794	GP-PG-N	0.814	GBN-BGP	0.815	GG-PG-N
0.887	GP-PB-N	0.790	GG-NG	0.836	GP-PB-N	0.796	GBN-BGG	0.669	GG-PB-N
0.842	GP-PG	0.790	GG-NN	0.812	GP-PG	0.811	GBN-BGN	0.769	GG-PG
0.834	GP-PB	0.790	GG-NB	0.812	GP-PB	0.812	GBN-BGB	0.768	GG-PB
0.786	P	0.790	P	0.734	P	0.768	P	0.780	P
0.897	GPN-PNN-N	0.789	GPG-NBN	0.809	GPN-PPP	0.792	GN-GN-N	0.824	GNP-PPN-N
0.784	B-N	0.786	B-N	0.749	B-N	0.748	B-N	0.783	B-N
0.825	B-P	0.774	B-P	0.749	B-P	0.811	B-P	0.762	B-P
0.769	B	0.774	B	0.720	B	0.745	B	0.758	B
0.768	B-G	0.773	B-G	0.767	B-G	0.751	B-G	0.809	B-G
0.834	PG	0.757	PG	0.766	PG	0.817	PG	0.830	PG
0.726	G	0.729	G	0.682	G	0.703	G	0.720	G
0.723	N	0.727	N	0.679	N	0.703	N	0.714	N
0.898	GB-PB-N	0.651	GB-NG	0.818	GB-PB-N	0.814	GB-BG	0.787	GB-PB-N

Tableau B-3 : Ajustements (coefficients de Nash-Sutcliffe) des prédictions sur la période P2, pour des modèles entraînés respectivement sur les périodes C, P1, P2, P3 et P4.

C : période complète (1993 – 2005) — Pi : période i — G : loi Gamma — N : loi normale — B : loi Béta — P : loi puissance — Fonctions de transfert en série avec les initiales juxtaposées (exemple : PN = loi puissance et loi normale en série) — Fonctions de transfert en parallèle avec les initiales séparées par un tiret (exemple : G-P = loi Gamma et loi puissance en parallèle).

P2_prédiction									
C_calibration		P1_calibration		P2_calibration		P3_calibration		P4_calibration	
0.818	GPB-PNP-N	0.778	GN-PP-N	0.856	GPB-PPP-N	0.733	GBP-GBP-N	0.684	GN-PP-N
0.831	G-P-N	0.778	G-P-N	0.856	G-P-N	0.776	G-P-N	0.693	G-P-N
0.831	G-N-N	0.778	G-N-N	0.856	G-N-N	0.674	G-N-N	0.693	G-N-N
0.831	GP-PP-N	0.751	GG-PP-N	0.856	GP-PP-N	0.748	GBN-GBP-N	0.707	GG-PP-N
0.816	GP-PN-N	0.736	GG-PG-N	0.856	GP-PN-N	0.733	GBN-GBG-N	0.709	GG-PN-N
0.810	GPP-PNP-N	0.723	GPP-NBP	0.856	GPP-PPP-N	0.718	GP-GP-N	0.750	GP-PP-N
0.823	GPP-PNG-N	0.720	GPP-NBG	0.856	GPP-PPN-N	0.708	GP-GG-N	0.744	GP-PN-N
0.805	GPN-PNG-N	0.687	GPG-NBG	0.855	GPN-PPN-N	0.714	GN-GG-N	0.706	GNP-PPG-N
0.824	GN-PP-N	0.683	GBP-PPP-N	0.854	GN-PP-N	0.705	GBG-GBP-N	0.664	GBP-PNP
0.812	GPN-PNP-N	0.753	GPG-NBP	0.853	GPN-PPP-N	0.785	GN-GP-N	0.754	GNP-PPP-N
0.823	GP-PG-N	0.756	GG-NP	0.852	GP-PG-N	0.733	GBN-BGP	0.736	GG-PG-N
0.827	GN-PN-N	0.731	GBP-PPG-N	0.850	GN-PN-N	0.710	GBG-GBG-N	0.627	GBP-PNG
0.825	G-G-N	0.772	G-G-N	0.850	G-G-N	0.717	G-G-N	0.691	G-G-N
0.804	GPP-PGP	0.722	GPN-NBP	0.850	GPP-PPG-N	0.746	GP-BP	0.688	GP-PG-N
0.823	GN-PG-N	0.684	GBN-PPP-N	0.850	GN-PG-N	0.689	GBG-BGP	0.765	GBN-PNP
0.813	GN-PN	0.732	GBP-PPB-N	0.845	GN-PN	0.688	GBG-GBB-N	0.713	GBP-PNB
0.817	GP-PN	0.749	GG-PB-N	0.845	GP-PN	0.739	GBN-GBB-N	0.750	GG-PN
0.806	GP-PP	0.684	GG-PN-N	0.845	GP-PP	0.744	GBN-GBN-N	0.751	GG-PP
0.819	GN-PP	0.677	GBP-PPN-N	0.845	GN-PP	0.697	GBG-GBN-N	0.755	GBP-PNN
0.756	GPP-PNB-N	0.724	GPP-NBB	0.845	GPP-PPN	0.717	GP-GB-N	0.769	GP-PN
0.815	GPP-PNN-N	0.720	GPP-NBN	0.845	GPP-PPP	0.716	GP-GN-N	0.769	GP-PP
0.819	G-N	0.759	G-N	0.845	G-N	0.719	G-N	0.769	G-N
0.812	G-P	0.795	G-P	0.845	G-P	0.762	G-P	0.769	G-P
0.828	GPN-PNN-N	0.752	GPG-NBN	0.845	GPN-PPP	0.702	GN-GN-N	0.765	GNP-PPN-N
0.805	GPN-PGP	0.732	GPB-NBP	0.845	GPN-PPG-N	0.712	GN-BP	0.754	GNN-PPP-N
0.803	GPN-PNB-N	0.685	GPG-NBB	0.845	GPN-PPN	0.715	GN-GB-N	0.704	GNP-PPB-N
0.819	GB-PN	0.683	GB-PB-N	0.845	GB-PN	0.728	GB-GB-N	0.768	GB-PN
0.760	GPB-PNB-N	0.713	GN-PB-N	0.844	GPB-PPN	0.732	GBP-GBB-N	0.768	GN-PN
0.814	N-P-N	0.769	N-P-N	0.844	N-P-N	0.776	N-P-N	0.667	N-P-N
0.812	N-N-N	0.769	N-N-N	0.843	N-N-N	0.786	N-N-N	0.669	N-N-N
0.793	GP-PB	0.754	GG-NB	0.843	GP-PB	0.734	GBN-BGB	0.744	GG-PB
0.820	P-P-N	0.760	P-P-N	0.842	P-P-N	0.786	P-P-N	0.770	P-P-N
0.818	P-N-N	0.755	P-N-N	0.841	P-N-N	0.719	P-N-N	0.693	P-N-N
0.794	N-B-N	0.766	N-B-N	0.841	N-B-N	0.727	N-B-N	0.731	N-B-N
0.816	G-G	0.759	G-G	0.840	G-G	0.709	G-G	0.764	G-G
0.803	GPP-PGN	0.718	GPN-NBN	0.840	GPP-PPG	0.720	GP-BN	0.764	GP-PG

0.810	GPN-PGN	0.731	GPB-NBN	0.840	GPN-PPG	0.726	GN-BN	0.743	GNN-PPN-N
0.796	GP-PG	0.747	GG-NN	0.840	GP-PG	0.737	GBN-BGN	0.720	GG-PG
0.816	GN-PG	0.688	GBN-PPN-N	0.840	GN-PG	0.693	GBG-BGN	0.658	GBN-PNN
0.804	N-G-N	0.758	N-G-N	0.834	N-G-N	0.772	N-G-N	0.661	N-G-N
0.812	P-G-N	0.750	P-G-N	0.833	P-G-N	0.723	P-G-N	0.758	P-G-N
0.802	N-N	0.774	N-N	0.832	N-N	0.784	N-N	0.751	N-N
0.802	N-P	0.786	N-P	0.832	N-P	0.747	N-P	0.751	N-P
0.808	P-P	0.777	P-P	0.831	P-P	0.770	P-P	0.756	P-P
0.780	N-B	0.758	N-B	0.831	N-B	0.711	N-B	0.756	N-B
0.778	GP-PB-N	0.749	GG-NG	0.831	GP-PB-N	0.729	GBN-BGG	0.622	GG-PB-N
0.806	P-N	0.796	P-N	0.830	P-N	0.709	P-N	0.758	P-N
0.809	GG-PN-N	0.645	GBG-PPG-N	0.830	GG-PN-N	0.738	GBB-GBG-N	0.628	GBG-PNG
0.801	G-B-N	0.773	G-B-N	0.828	G-B-N	0.745	G-B-N	0.740	G-B-N
0.742	P-B	0.763	P-B	0.828	P-B	0.725	P-B	0.734	P-B
0.797	G-B	0.767	G-B	0.828	G-B	0.709	G-B	0.759	G-B
0.796	N-G	0.769	N-G	0.824	N-G	0.704	N-G	0.745	N-G
0.796	GPG-PNG-N	0.749	GP-PG-N	0.823	GPG-PPN-N	0.683	GG-GG-N	0.497	GNG-PPG-N
0.801	P-G	0.728	P-G	0.823	P-G	0.712	P-G	0.750	P-G
0.790	GPP-PGB	0.722	GPN-NBB	0.822	GPP-PPB	0.715	GP-BB	0.758	GP-PB
0.747	GPG-PNN-N	0.747	GP-PN-N	0.822	GPG-PPP	0.682	GG-GN-N	0.541	GNG-PPN-N
0.796	GG-PP	0.674	GBG-PPN-N	0.822	GG-PP	0.739	GBB-GBN-N	0.656	GBG-PNN
0.795	GG-PN	0.618	GBG-PPB-N	0.822	GG-PN	0.739	GBB-GBB-N	0.682	GBG-PNB
0.773	GPG-PGP	0.768	GP-NP	0.817	GPG-PPG-N	0.737	GG-BP	0.529	GNB-PPP-N
0.798	GG-PG-N	0.684	GBB-PPP-N	0.815	GG-PG-N	0.731	GBB-BGP	0.658	GBB-PNP
0.711	GPG-PGB	0.722	GP-NB	0.814	GPG-PPB	0.695	GG-BB	0.747	GNB-PPB-N
0.824	B-B-N	0.658	B-B-N	0.810	B-B-N	0.713	B-B-N	0.746	B-B-N
0.776	B-N-N	0.729	B-N-N	0.810	B-N-N	0.739	B-N-N	0.623	B-N-N
0.776	B-P-N	0.729	B-P-N	0.810	B-P-N	0.797	B-P-N	0.587	B-P-N
0.796	GPP-PGG	0.724	GPN-NBG	0.810	GPP-PPB-N	0.720	GP-BG	0.738	GP-PB-N
0.749	GPN-PGB	0.738	GPB-NBB	0.808	GPN-PPB	0.718	GN-BB	0.695	GNN-PPB-N
0.796	GG-PG	0.684	GBB-PPN-N	0.807	GG-PG	0.731	GBB-BGN	0.666	GBB-PNN
0.781	GPG-PGN	0.761	GP-NN	0.807	GPG-PPG	0.696	GG-BN	0.523	GNB-PPN-N
0.810	GPG-PNP-N	0.779	GP-PP-N	0.806	GPG-PPP-N	0.780	GG-GP-N	0.732	GNG-PPP-N
0.810	GG-PP-N	0.683	GBG-PPP-N	0.806	GG-PP-N	0.733	GBB-GBP-N	0.661	GBG-PNP
0.821	GB-PP-N	0.758	GB-PP-N	0.806	GB-PP-N	0.806	GB-GP-N	0.506	GB-PP-N
0.696	GPB-PNG-N	0.678	GN-PG-N	0.806	GPB-PPN-N	0.736	GBP-GBG-N	0.766	GN-PN-N
0.754	GN-PB-N	0.643	GBN-PPG-N	0.804	GN-PB-N	0.701	GBG-BGG	0.766	GBN-PNG
0.631	GPB-PGG	0.766	GN-NG	0.800	GPB-PPB-N	0.731	GBP-BGG	0.656	GN-PB-N
0.757	GB-PB-N	0.726	GB-NG	0.800	GB-PB-N	0.734	GB-BG	0.644	GB-PB-N
0.799	GPN-PGG	0.733	GPB-NBG	0.800	GPN-PPB-N	0.704	GN-BG	0.663	GNN-PPG-N
0.753	B-G	0.732	B-G	0.800	B-G	0.731	B-G	0.750	B-G
0.758	GPB-PGB	0.771	GN-NB	0.799	GPB-PPB	0.731	GBP-BGB	0.757	GN-PB
0.767	B-G-N	0.718	B-G-N	0.799	B-G-N	0.791	B-G-N	0.616	B-G-N
0.819	GB-PN-N	0.771	GB-PG-N	0.799	GB-PN-N	0.784	GB-GG-N	0.770	GB-PN-N

0.803	GN-PB	0.749	GBN-PPB-N	0.797	GN-PB	0.695	GBG-BGB	0.745	GBN-PNB
0.819	GB-PB	0.730	GB-NB	0.797	GB-PB	0.736	GB-BB	0.719	GB-PB
0.772	B-B	0.769	B-B	0.793	B-B	0.689	B-B	0.685	B-B
0.808	B-P	0.739	B-P	0.791	B-P	0.793	B-P	0.711	B-P
0.771	B-N	0.759	B-N	0.791	B-N	0.729	B-N	0.739	B-N
0.749	P-B-N	0.675	P-B-N	0.782	P-B-N	0.723	P-B-N	0.747	P-B-N
0.773	GPG-PGG	0.751	GP-NG	0.778	GPG-PPB-N	0.666	GG-BG	0.368	GNB-PPG-N
0.757	GG-PB	0.688	GBB-PPB-N	0.775	GG-PB	0.731	GBB-BGB	0.726	GBB-PNB
0.818	GPB-PNN-N	0.776	GN-PN-N	0.772	GPB-PPP	0.741	GBP-GBN-N	0.769	GN-PP
0.822	GB-PP	0.684	GB-PN-N	0.772	GB-PP	0.805	GB-GN-N	0.768	GB-PP
0.754	GPG-PNB-N	0.684	GP-PB-N	0.772	GPG-PPN	0.685	GG-GB-N	0.635	GNG-PPB-N
0.816	GB-PG-N	0.840	GB-NP	0.771	GB-PG-N	0.731	GB-BP	0.495	GB-PG-N
0.742	GG-PB-N	0.646	GBB-PPG-N	0.771	GG-PB-N	0.731	GBB-BGG	0.623	GBB-PNG
0.818	GPB-PGP	0.759	GN-NP	0.758	GPB-PPG-N	0.728	GBP-BGP	0.684	GN-PG-N
0.704	PGB	0.679	PN	0.743	PGB	0.696	PGB	0.655	PGB
0.696	PG	0.682	PG	0.741	PG	0.670	PG	0.654	PG
0.694	PGN	0.678	PNG	0.739	PGN	0.672	PGN	0.650	PGN
0.699	PN	0.679	PNP	0.739	PN	0.659	PN	0.656	PN
0.692	GB-PG	0.795	GB-NN	0.737	GB-PG	0.736	GB-BN	0.625	GB-PG
0.669	GPB-PGN	0.758	GN-NN	0.737	GPB-PPG	0.721	GBP-BGN	0.765	GN-PG
0.692	PB	0.679	PB	0.731	PB	0.674	PB	0.650	PB
0.688	PP	0.673	PP	0.726	PP	0.677	PP	0.643	PP
0.674	P	0.660	P	0.710	P	0.657	P	0.630	P
0.654	PGG	0.685	PNB	0.699	PGG	0.588	PGG	0.617	PGG
0.652	B	0.638	B	0.688	B	0.621	B	0.611	B
0.692	PGP	0.677	PNN	0.675	PGP	0.667	PGP	0.653	PGP
0.643	G	0.622	G	0.675	G	0.629	G	0.571	G
0.639	N	0.617	N	0.667	N	0.621	N	0.568	N

Tableau B-4 : Ajustements (coefficients de Nash-Sutcliffe) des prédictions sur la période P3, pour des modèles entraînés respectivement sur les périodes C, P1, P2, P3 et P4.

C : période complète (1993 – 2005) — P_i : période i — G : loi Gamma — N : loi normale — B : loi Béta — P : loi puissance — Fonctions de transfert en série avec les initiales juxtaposées (exemple : PN = loi puissance et loi normale en série) — Fonctions de transfert en parallèle avec les initiales séparées par un tiret (exemple : G-P = loi Gamma et loi puissance en parallèle).

P3_prédiction									
C_calibration		P1_calibration		P2_calibration		P3_calibration		P4_calibration	
0.793	GB-PN	0.615	GB-PB-N	0.746	GB-PN	0.834	GB-GB-N	0.779	GB-PN
0.775	GG-PN	0.532	GBG-PPB-N	0.729	GG-PN	0.832	GBB-GBB-N	0.762	GBG-PNB
0.783	GG-PB	0.618	GBB-PPB-N	0.713	GG-PB	0.832	GBB-BGB	0.784	GBB-PNB
0.810	GPB-PGB	0.759	GN-NB	0.737	GPB-PPB	0.832	GBP-BGB	0.778	GN-PB
0.800	GB-PB-N	0.571	GB-NG	0.724	GB-PB-N	0.831	GB-BG	0.654	GB-PB-N
0.505	GB-PG	0.765	GB-NN	0.374	GB-PG	0.831	GB-BN	0.405	GB-PG
0.767	GG-PN-N	0.232	GBG-PPG-N	0.722	GG-PN-N	0.830	GBB-GBG-N	0.406	GBG-PNG
0.804	GPP-PGN	0.776	GPN-NBN	0.755	GPP-PPG	0.830	GP-BN	0.774	GP-PG
0.790	GP-PN-N	0.553	GG-PG-N	0.730	GP-PN-N	0.830	GBN-GBG-N	0.649	GG-PN-N
0.776	GG-PP	0.232	GBG-PPN-N	0.729	GG-PP	0.830	GBB-GBN-N	0.413	GBG-PNN
0.775	GPP-PNG-N	0.775	GPP-NBG	0.730	GPP-PPN-N	0.830	GP-GG-N	0.720	GP-PN-N
0.785	GPB-PNN-N	0.579	GN-PN-N	0.380	GPB-PPP	0.830	GBP-GBN-N	0.775	GN-PP
0.765	GG-PP-N	0.247	GBG-PPP-N	0.400	GG-PP-N	0.830	GBB-GBP-N	0.412	GBG-PNP
0.528	GPB-PGN	0.759	GN-NN	0.381	GPB-PPG	0.830	GBP-BGN	0.774	GN-PG
0.760	GG-PG-N	0.251	GBB-PPP-N	0.720	GG-PG-N	0.829	GBB-BGP	0.407	GBB-PNP
0.802	GP-PP	0.249	GG-PN-N	0.750	GP-PP	0.829	GBN-GBN-N	0.761	GG-PP
0.798	GPP-PGG	0.783	GPN-NBG	0.750	GPP-PPB-N	0.829	GP-BG	0.764	GP-PB-N
0.794	GPB-PNB-N	0.594	GN-PB-N	0.756	GPB-PPN	0.829	GBP-GBB-N	0.775	GN-PN
0.792	GPB-PGP	0.758	GN-NP	0.426	GPB-PPG-N	0.828	GBP-BGP	0.662	GN-PG-N
0.792	GB-PB	0.776	GB-NB	0.723	GB-PB	0.828	GB-BB	0.788	GB-PB
0.798	GPP-PGB	0.779	GPN-NBB	0.752	GPP-PPB	0.828	GP-BB	0.779	GP-PB
0.783	GB-PG-N	0.774	GB-NP	0.402	GB-PG-N	0.828	GB-BP	0.025	GB-PG-N
0.760	GG-PB-N	0.238	GBB-PPG-N	0.695	GG-PB-N	0.828	GBB-BGG	0.395	GBB-PNG
0.728	GG-PG	0.251	GBB-PPN-N	0.729	GG-PG	0.828	GBB-BGN	0.411	GBB-PNN
0.790	GPP-PNP-N	0.779	GPP-NBP	0.730	GPP-PPP-N	0.828	GP-GP-N	0.779	GP-PP-N
0.572	GPB-PNG-N	0.554	GN-PG-N	0.408	GPB-PPN-N	0.827	GBP-GBG-N	0.784	GN-PN-N
0.781	GP-PG-N	0.755	GG-NP	0.766	GP-PG-N	0.826	GBN-BGP	0.753	GG-PG-N
0.788	GPB-PNP-N	0.578	GN-PP-N	0.733	GPB-PPP-N	0.826	GBP-GBP-N	0.658	GN-PP-N
0.776	GP-PP-N	0.555	GG-PP-N	0.730	GP-PP-N	0.826	GBN-GBP-N	0.649	GG-PP-N
0.543	GPB-PGG	0.760	GN-NG	0.728	GPB-PPB-N	0.825	GBP-BGG	0.743	GN-PB-N
0.805	GP-PB	0.757	GG-NB	0.753	GP-PB	0.823	GBN-BGB	0.758	GG-PB
0.786	GPN-PNG-N	0.766	GPG-NBG	0.731	GPN-PPN-N	0.823	GN-GG-N	0.718	GNP-PPG-N
0.789	GPP-PNN-N	0.776	GPP-NBN	0.750	GPP-PPP	0.823	GP-GN-N	0.775	GP-PP
0.776	G-G-N	0.580	G-G-N	0.732	G-G-N	0.823	G-G-N	0.679	G-G-N
0.810	GP-PG	0.755	GG-NN	0.753	GP-PG	0.823	GBN-BGN	0.749	GG-PG
0.795	GP-PN	0.562	GG-PB-N	0.750	GP-PN	0.822	GBN-GBB-N	0.761	GG-PN

0.776	G-P-N	0.578	G-P-N	0.730	G-P-N	0.822	G-P-N	0.663	G-P-N
0.777	GPN-PNN-N	0.758	GPG-NBN	0.751	GPN-PPP	0.820	GN-GN-N	0.779	GNP-PPN-N
0.793	GPN-PNP-N	0.769	GPG-NBP	0.726	GPN-PPP-N	0.819	GN-GP-N	0.786	GNP-PPP-N
0.800	GPP-PNB-N	0.778	GPP-NBB	0.750	GPP-PPN	0.819	GP-GB-N	0.775	GP-PN
0.789	G-B-N	0.579	G-B-N	0.747	G-B-N	0.818	G-B-N	0.779	G-B-N
0.798	GPN-PGG	0.775	GPB-NBG	0.702	GPN-PPB-N	0.818	GN-BG	0.664	GNN-PPG-N
0.800	G-B	0.759	G-B	0.755	G-B	0.817	G-B	0.776	G-B
0.789	G-G	0.758	G-G	0.755	G-G	0.817	G-G	0.774	G-G
0.788	G-N	0.758	G-N	0.750	G-N	0.817	G-N	0.775	G-N
0.800	GPP-PGP	0.777	GPN-NBP	0.731	GPP-PPG-N	0.817	GP-BP	0.664	GP-PG-N
0.810	GPN-PGB	0.780	GPB-NBB	0.750	GPN-PPB	0.817	GN-BB	0.778	GNN-PPB-N
0.787	GPN-PGN	0.780	GPB-NBN	0.755	GPN-PPG	0.816	GN-BN	0.752	GNN-PPN-N
0.779	GP-PB-N	0.755	GG-NG	0.746	GP-PB-N	0.816	GBN-BGG	0.607	GG-PB-N
0.801	G-P	0.771	G-P	0.750	G-P	0.816	G-P	0.775	G-P
0.776	G-N-N	0.578	G-N-N	0.730	G-N-N	0.815	G-N-N	0.662	G-N-N
0.785	GB-PN-N	0.576	GB-PG-N	0.409	GB-PN-N	0.804	GB-GG-N	0.777	GB-PN-N
0.777	GN-PG-N	0.248	GBN-PPP-N	0.733	GN-PG-N	0.803	GBG-BGP	0.780	GBN-PNP
0.790	GN-PP	0.232	GBP-PPN-N	0.750	GN-PP	0.803	GBG-GBN-N	0.774	GBP-PNN
0.785	GN-PP-N	0.252	GBP-PPP-N	0.729	GN-PP-N	0.803	GBG-GBP-N	0.415	GBP-PNP
0.790	GN-PG	0.253	GBN-PPN-N	0.756	GN-PG	0.802	GBG-BGN	0.410	GBN-PNN
0.762	GN-PB-N	0.240	GBN-PPG-N	0.714	GN-PB-N	0.801	GBG-BGG	0.780	GBN-PNG
0.761	GPG-PNG-N	0.603	GP-PG-N	0.719	GPG-PPN-N	0.801	GG-GG-N	0.029	GNG-PPG-N
0.768	GPG-PNB-N	0.616	GP-PB-N	0.389	GPG-PPN	0.801	GG-GB-N	0.676	GNG-PPB-N
0.572	GPG-PNN-N	0.602	GP-PN-N	0.730	GPG-PPP	0.800	GG-GN-N	0.039	GNG-PPN-N
0.799	GN-PB	0.591	GBN-PPB-N	0.708	GN-PB	0.800	GBG-BGB	0.783	GBN-PNB
0.794	GB-PP	0.248	GB-PN-N	0.380	GB-PP	0.800	GB-GN-N	0.780	GB-PP
0.783	GB-PP-N	0.600	GB-PP-N	0.397	GB-PP-N	0.799	GB-GP-N	0.039	GB-PP-N
0.777	GN-PN-N	0.607	GBP-PPG-N	0.742	GN-PN-N	0.798	GBG-GBG-N	0.402	GBP-PNG
0.796	GN-PN	0.608	GBP-PPB-N	0.751	GN-PN	0.796	GBG-GBB-N	0.786	GBP-PNB
0.753	N-P-N	0.544	N-P-N	0.720	N-P-N	0.795	N-P-N	0.627	N-P-N
0.754	N-G-N	0.542	N-G-N	0.704	N-G-N	0.793	N-G-N	0.628	N-G-N
0.775	GPN-PNB-N	0.763	GPG-NBB	0.753	GPN-PPN	0.793	GN-GB-N	0.780	GNP-PPB-N
0.801	GPN-PGP	0.773	GPB-NBP	0.728	GPN-PPG-N	0.793	GN-BP	0.757	GNN-PPP-N
0.779	B-P	0.677	B-P	0.650	B-P	0.792	B-P	0.682	B-P
0.782	GPG-PGP	0.782	GP-NP	0.723	GPG-PPG-N	0.791	GG-BP	0.039	GNB-PPP-N
0.703	B-P-N	0.496	B-P-N	0.643	B-P-N	0.791	B-P-N	0.459	B-P-N
0.702	B-G-N	0.497	B-G-N	0.644	B-G-N	0.790	B-G-N	0.517	B-G-N
0.739	P-G-N	0.545	P-G-N	0.698	P-G-N	0.789	P-G-N	0.721	P-G-N
0.740	P-N-N	0.548	P-N-N	0.698	P-N-N	0.789	P-N-N	0.626	P-N-N
0.768	N-P	0.749	N-P	0.716	N-P	0.789	N-P	0.745	N-P
0.771	GPG-PGN	0.759	GP-NN	0.728	GPG-PPG	0.788	GG-BN	0.024	GNB-PPN-N
0.768	GPG-PGG	0.769	GP-NG	0.715	GPG-PPB-N	0.787	GG-BG	0.148	GNB-PPG-N
0.790	GPG-PGB	0.778	GP-NB	0.736	GPG-PPB	0.787	GG-BB	0.785	GNB-PPB-N
0.765	GPG-PNP-N	0.577	GP-PP-N	0.401	GPG-PPP-N	0.785	GG-GP-N	0.756	GNG-PPP-N

0.755	N-B-N	0.542	N-B-N	0.719	N-B-N	0.783	N-B-N	0.737	N-B-N
0.749	P-B-N	0.556	P-B-N	0.670	P-B-N	0.781	P-B-N	0.736	P-B-N
0.756	N-N-N	0.546	N-N-N	0.699	N-N-N	0.780	N-N-N	0.628	N-N-N
0.768	N-N	0.742	N-N	0.716	N-N	0.778	N-N	0.745	N-N
0.768	N-G	0.741	N-G	0.717	N-G	0.778	N-G	0.742	N-G
0.762	N-B	0.744	N-B	0.718	N-B	0.776	N-B	0.734	N-B
0.744	P-G	0.732	P-G	0.714	P-G	0.776	P-G	0.729	P-G
0.745	P-N	0.730	P-N	0.712	P-N	0.774	P-N	0.732	P-N
0.753	P-B	0.738	P-B	0.714	P-B	0.768	P-B	0.738	P-B
0.747	P-P	0.741	P-P	0.713	P-P	0.768	P-P	0.731	P-P
0.741	P-P-N	0.546	P-P-N	0.696	P-P-N	0.766	P-P-N	0.731	P-P-N
0.785	B-B	0.772	B-B	0.691	B-B	0.764	B-B	0.711	B-B
0.746	B-B-N	0.532	B-B-N	0.726	B-B-N	0.763	B-B-N	0.759	B-B-N
0.736	PGN	0.721	PNG	0.655	PGN	0.753	PGN	0.712	PGN
0.736	PG	0.640	PG	0.653	PG	0.753	PG	0.716	PG
0.739	PGP	0.718	PNN	0.544	PGP	0.753	PGP	0.718	PGP
0.703	B-N-N	0.496	B-N-N	0.643	B-N-N	0.752	B-N-N	0.511	B-N-N
0.727	PN	0.711	PNP	0.653	PN	0.745	PN	0.706	PN
0.733	PGB	0.707	PN	0.662	PGB	0.742	PGB	0.718	PGB
0.708	B-N	0.702	B-N	0.650	B-N	0.720	B-N	0.707	B-N
0.694	B-G	0.675	B-G	0.676	B-G	0.719	B-G	0.757	B-G
0.695	PB	0.680	PB	0.624	PB	0.703	PB	0.671	PB
0.683	PP	0.668	PP	0.618	PP	0.693	PP	0.666	PP
0.657	P	0.641	P	0.591	P	0.667	P	0.638	P
0.641	B	0.624	B	0.572	B	0.657	B	0.613	B
0.594	G	0.582	G	0.544	G	0.603	G	0.570	G
0.583	N	0.563	N	0.527	N	0.595	N	0.552	N
0.718	PGG	0.713	PNB	0.624	PGG	0.571	PGG	0.694	PGG

Tableau B-5 : Ajustements (coefficients de Nash-Sutcliffe) des prédictions sur la période P4, pour des modèles entraînés respectivement sur les périodes C, P1, P2, P3 et P4.

C : période complète (1993 – 2005) — P_i : période i — G : loi Gamma — N : loi normale — B : loi Béta — P : loi puissance — Fonctions de transfert en série avec les initiales juxtaposées (exemple : PN = loi puissance et loi normale en série) — Fonctions de transfert en parallèle avec les initiales séparées par un tiret (exemple : G-P = loi Gamma et loi puissance en parallèle).

P4_prédiction									
C_calibration		P1_calibration		P2_calibration		P3_calibration		P4_calibration	
0.832	G-P-N	0.688	G-P-N	0.777	G-P-N	0.826	G-P-N	0.918	G-P-N
0.832	G-N-N	0.686	G-N-N	0.777	G-N-N	0.821	G-N-N	0.918	G-N-N
0.833	GPB-PNP-N	0.687	GN-PP-N	0.777	GPB-PPP-N	0.821	GBP-GBP-N	0.917	GN-PP-N
0.852	GPP-PGP	0.848	GPN-NBP	0.775	GPP-PPG-N	0.821	GP-BP	0.915	GP-PG-N
0.851	GPB-PGP	0.833	GN-NP	0.525	GPB-PPG-N	0.814	GBP-BGP	0.914	GN-PG-N
0.830	G-G-N	0.691	G-G-N	0.776	G-G-N	0.806	G-G-N	0.914	G-G-N
0.830	GPP-PNG-N	0.849	GPP-NBG	0.777	GPP-PPN-N	0.810	GP-GG-N	0.907	GP-PN-N
0.848	GPN-PGG	0.849	GPB-NBG	0.737	GPN-PPB-N	0.809	GN-BG	0.906	GNN-PPG-N
0.806	N-P-N	0.664	N-P-N	0.769	N-P-N	0.776	N-P-N	0.900	N-P-N
0.804	N-N-N	0.665	N-N-N	0.751	N-N-N	0.808	N-N-N	0.899	N-N-N
0.828	GPN-PNG-N	0.781	GPG-NBG	0.778	GPN-PPN-N	0.805	GN-GG-N	0.898	GNP-PPG-N
0.802	N-G-N	0.666	N-G-N	0.731	N-G-N	0.770	N-G-N	0.894	N-G-N
0.821	GB-PB-N	0.539	GB-NG	0.764	GB-PB-N	0.817	GB-BG	0.888	GB-PB-N
0.738	B-P-N	0.608	B-P-N	0.682	B-P-N	0.821	B-P-N	0.883	B-P-N
0.849	GPN-PGP	0.849	GPB-NBP	0.764	GPN-PPG-N	0.797	GN-BP	0.882	GNN-PPP-N
0.804	P-N-N	0.669	P-N-N	0.749	P-N-N	0.761	P-N-N	0.880	P-N-N
0.832	GP-PP-N	0.638	GG-PP-N	0.777	GP-PP-N	0.813	GBN-GBP-N	0.878	GG-PP-N
0.851	GPP-PGG	0.847	GPN-NBG	0.758	GPP-PPB-N	0.813	GP-BG	0.878	GP-PB-N
0.837	GP-PN-N	0.636	GG-PG-N	0.777	GP-PN-N	0.815	GBN-GBG-N	0.876	GG-PN-N
0.842	GPN-PGN	0.850	GPB-NBN	0.793	GPN-PPG	0.813	GN-BN	0.874	GNN-PPN-N
0.836	GPP-PNP-N	0.848	GPP-NBP	0.777	GPP-PPP-N	0.819	GP-GP-N	0.873	GP-PP-N
0.833	GB-PN-N	0.696	GB-PG-N	0.481	GB-PN-N	0.820	GB-GG-N	0.870	GB-PN-N
0.832	GPN-PNP-N	0.777	GPG-NBP	0.770	GPN-PPP-N	0.796	GN-GP-N	0.869	GNP-PPP-N
0.853	GB-PN	0.709	GB-PB-N	0.796	GB-PN	0.807	GB-GB-N	0.869	GB-PN
0.850	GB-PP	0.402	GB-PN-N	0.492	GB-PP	0.844	GB-GN-N	0.868	GB-PP
0.830	GN-PG-N	0.403	GBN-PPP-N	0.776	GN-PG-N	0.754	GBG-BGP	0.867	GBN-PNP
0.848	GN-PP	0.396	GBP-PPN-N	0.793	GN-PP	0.751	GBG-GBN-N	0.867	GBP-PNN
0.834	G-B-N	0.694	G-B-N	0.791	G-B-N	0.825	G-B-N	0.867	G-B-N
0.831	GPN-PNN-N	0.772	GPG-NBN	0.793	GPN-PPP	0.813	GN-GN-N	0.867	GNP-PPN-N
0.832	GPB-PNN-N	0.687	GN-PN-N	0.493	GPB-PPP	0.821	GBP-GBN-N	0.866	GN-PP
0.837	GPP-PNN-N	0.849	GPP-NBN	0.793	GPP-PPP	0.806	GP-GN-N	0.866	GP-PP
0.828	GPP-PNB-N	0.848	GPP-NBB	0.793	GPP-PPN	0.818	GP-GB-N	0.866	GP-PN
0.849	G-N	0.834	G-N	0.793	G-N	0.814	G-N	0.866	G-N
0.850	G-P	0.836	G-P	0.793	G-P	0.830	G-P	0.866	G-P
0.824	GPB-PNB-N	0.682	GN-PB-N	0.792	GPB-PPN	0.810	GBP-GBB-N	0.866	GN-PN
0.779	GPG-PGB	0.845	GP-NB	0.750	GPG-PPB	0.755	GG-BB	0.866	GNB-PPB-N

0.845	GPB-PGB	0.835	GN-NB	0.775	GPB-PPB	0.814	GBP-BGB	0.865	GN-PB
0.850	G-B	0.834	G-B	0.806	G-B	0.812	G-B	0.865	G-B
0.848	GPP-PGB	0.848	GPN-NBB	0.792	GPP-PPB	0.812	GP-BB	0.865	GP-PB
0.850	GN-PB	0.708	GBN-PPB-N	0.746	GN-PB	0.760	GBG-BGB	0.865	GBN-PNB
0.852	GPP-PGN	0.838	GPN-NBN	0.792	GPP-PPG	0.813	GP-BN	0.863	GP-PG
0.846	G-G	0.833	G-G	0.791	G-G	0.813	G-G	0.863	G-G
0.515	GPB-PGN	0.831	GN-NN	0.457	GPB-PPG	0.813	GBP-BGN	0.863	GN-PG
0.788	GG-PB	0.717	GBB-PPB-N	0.734	GG-PB	0.815	GBB-BGB	0.863	GBB-PNB
0.550	GPB-PNG-N	0.648	GN-PG-N	0.478	GPB-PPN-N	0.816	GBP-GBG-N	0.863	GN-PN-N
0.792	GN-PB-N	0.394	GBN-PPG-N	0.744	GN-PB-N	0.762	GBG-BGG	0.861	GBN-PNG
0.828	GPN-PNB-N	0.780	GPG-NBB	0.797	GPN-PPN	0.810	GN-GB-N	0.861	GPN-PPB-N
0.853	GB-PB	0.844	GB-NB	0.759	GB-PB	0.816	GB-BB	0.860	GB-PB
0.848	GN-PN	0.704	GBP-PPB-N	0.793	GN-PN	0.731	GBG-GBB-N	0.858	GBP-PNB
0.812	B-B-N	0.649	B-B-N	0.767	B-B-N	0.756	B-B-N	0.853	B-B-N
0.847	GPN-PGB	0.849	GPB-NBB	0.773	GPN-PPB	0.812	GN-BB	0.850	GNN-PPB-N
0.745	B-G	0.747	B-G	0.729	B-G	0.734	B-G	0.844	B-G
0.824	PG	0.765	PG	0.734	PG	0.801	PG	0.843	PG
0.825	PGP	0.827	PNN	0.653	PGP	0.796	PGP	0.843	PGP
0.805	P-P-N	0.669	P-P-N	0.752	P-P-N	0.792	P-P-N	0.841	P-P-N
0.824	PGN	0.829	PNG	0.732	PGN	0.805	PGN	0.841	PGN
0.827	PGB	0.822	PN	0.731	PGB	0.814	PGB	0.840	PGB
0.817	N-P	0.810	N-P	0.765	N-P	0.787	N-P	0.839	N-P
0.817	N-N	0.810	N-N	0.765	N-N	0.804	N-N	0.839	N-N
0.808	N-B-N	0.665	N-B-N	0.767	N-B-N	0.791	N-B-N	0.836	N-B-N
0.802	P-G-N	0.671	P-G-N	0.751	P-G-N	0.769	P-G-N	0.836	P-G-N
0.738	B-N-N	0.608	B-N-N	0.682	B-N-N	0.729	B-N-N	0.835	B-N-N
0.820	PN	0.826	PNP	0.731	PN	0.779	PN	0.835	PN
0.787	P-B-N	0.670	P-B-N	0.719	P-B-N	0.767	P-B-N	0.835	P-B-N
0.815	N-B	0.818	N-B	0.764	N-B	0.781	N-B	0.835	N-B
0.819	P-P	0.815	P-P	0.764	P-P	0.800	P-P	0.834	P-P
0.811	N-G	0.810	N-G	0.762	N-G	0.776	N-G	0.834	N-G
0.819	P-N	0.813	P-N	0.763	P-N	0.766	P-N	0.833	P-N
0.481	GPB-PGG	0.835	GN-NG	0.766	GPB-PPB-N	0.819	GBP-BGG	0.831	GN-PB-N
0.813	P-G	0.804	P-G	0.762	P-G	0.778	P-G	0.829	P-G
0.736	B-G-N	0.612	B-G-N	0.681	B-G-N	0.812	B-G-N	0.829	B-G-N
0.801	P-B	0.813	P-B	0.765	P-B	0.779	P-B	0.827	P-B
0.783	GPG-PNB-N	0.712	GP-PB-N	0.493	GPG-PPN	0.760	GG-GB-N	0.825	GNG-PPB-N
0.814	GP-PB-N	0.773	GG-NG	0.798	GP-PB-N	0.803	GBN-BGG	0.823	GG-PB-N
0.829	GP-PG-N	0.770	GG-NP	0.711	GP-PG-N	0.813	GBN-BGP	0.821	GG-PG-N
0.796	GPG-PNP-N	0.687	GP-PP-N	0.480	GPG-PPP-N	0.798	GG-GP-N	0.816	GNG-PPP-N
0.853	GP-PP	0.403	GG-PN-N	0.793	GP-PP	0.817	GBN-GBN-N	0.814	GG-PP
0.850	GP-PN	0.640	GG-PB-N	0.793	GP-PN	0.816	GBN-GBB-N	0.814	GG-PN
0.848	GP-PB	0.772	GG-NB	0.794	GP-PB	0.820	GBN-BGB	0.813	GG-PB
0.780	B-N	0.780	B-N	0.689	B-N	0.730	B-N	0.810	B-N

0.796	PP	0.797	PP	0.705	PP	0.779	PP	0.810	PP
0.799	PB	0.804	PB	0.708	PB	0.777	PB	0.806	PB
0.854	GP-PG	0.774	GG-NN	0.790	GP-PG	0.816	GBN-BGN	0.804	GG-PG
0.827	B-B	0.818	B-B	0.742	B-B	0.765	B-B	0.799	B-B
0.796	GG-PN	0.609	GBG-PPB-N	0.746	GG-PN	0.820	GBB-GBB-N	0.796	GBG-PNB
0.564	GPG-PNN-N	0.708	GP-PN-N	0.747	GPG-PPP	0.761	GG-GN-N	0.793	GNG-PPN-N
0.792	GPG-PGP	0.854	GP-NP	0.735	GPG-PPG-N	0.775	GG-BP	0.792	GNB-PPP-N
0.835	GB-PP-N	0.706	GB-PP-N	0.478	GB-PP-N	0.840	GB-GP-N	0.791	GB-PP-N
0.783	GPG-PGN	0.834	GP-NN	0.736	GPG-PPG	0.752	GG-BN	0.790	GNB-PPN-N
0.768	P	0.772	P	0.682	P	0.749	P	0.783	P
0.759	PGG	0.830	PNB	0.666	PGG	0.654	PGG	0.776	PGG
0.833	GB-PG-N	0.819	GB-NP	0.456	GB-PG-N	0.814	GB-BP	0.770	GB-PG-N
0.786	GPG-PNG-N	0.715	GP-PG-N	0.743	GPG-PPN-N	0.760	GG-GG-N	0.769	GNG-PPG-N
0.829	B-P	0.751	B-P	0.689	B-P	0.817	B-P	0.764	B-P
0.740	G	0.749	G	0.653	G	0.711	G	0.759	G
0.738	B	0.741	B	0.658	B	0.701	B	0.757	B
0.723	N	0.732	N	0.636	N	0.702	N	0.740	N
0.783	GPG-PGG	0.848	GP-NG	0.726	GPG-PPB-N	0.750	GG-BG	0.716	GNB-PPG-N
0.787	GG-PG-N	0.404	GBB-PPP-N	0.729	GG-PG-N	0.817	GBB-BGP	0.621	GBB-PNP
0.843	GN-PG	0.406	GBN-PPN-N	0.794	GN-PG	0.761	GBG-BGN	0.620	GBN-PNN
0.794	GG-PP-N	0.402	GBG-PPP-N	0.477	GG-PP-N	0.813	GBB-GBP-N	0.620	GBG-PNP
0.807	GG-PG	0.404	GBB-PPN-N	0.737	GG-PG	0.818	GBB-BGN	0.620	GBB-PNN
0.834	GN-PP-N	0.404	GBP-PPP-N	0.775	GN-PP-N	0.760	GBG-GBP-N	0.620	GBP-PNP
0.796	GG-PP	0.392	GBG-PPN-N	0.746	GG-PP	0.806	GBB-GBN-N	0.616	GBG-PNN
0.758	GG-PB-N	0.390	GBB-PPG-N	0.706	GG-PB-N	0.817	GBB-BGG	0.593	GBB-PNG
0.793	GG-PN-N	0.388	GBG-PPG-N	0.739	GG-PN-N	0.818	GBB-GBG-N	0.592	GBG-PNG
0.526	GB-PG	0.850	GB-NN	0.455	GB-PG	0.818	GB-BN	0.591	GB-PG
0.828	GN-PN-N	0.705	GBP-PPG-N	0.785	GN-PN-N	0.763	GBG-GBG-N	0.591	GBP-PNG

RÉFÉRENCES BIBLIOGRAPHIQUES

- Ambroise, B. (1999). Genèse des débits dans les petits bassins versants ruraux en milieu tempéré 2-Modélisation systémique et dynamique. *Revue Des Sciences de L'eau*.
- Andréassian, V., Le Moine, N., Perrin, C., Ramos, M. H., Oudin, L., Mathevet, T., ... Berthet, L. (2012). All that glitters is not gold: the case of calibrating hydrological models. *Hydrological Processes*, 26, p. 2206 – p. 2210. doi:10.1002/hyp.9264
- Bellman, R. (1961). *Adaptive Control Processes: A Guided Tour*. Princeton University Press. Retrieved from <http://books.google.fr/books?id=POAmAAAAMAAJ>
- Besbes, M., & de Marsily, G. (1984). From infiltration to recharge: use of a parametric transfer function. *Journal of Hydrology*, 74(3), 271 – 293.
- Bessiere, H. (2008). *Assimilation de données variationnelle pour la modélisation hydrologique distribuée des crues à cinétique rapide*. Sciences-New York.
- Beven, K. J., & Kirkby, M. J. (1979). A physically based , variable contributing area model of basin hydrology / Un modèle à base physique de zone d'appel variable de l'hydrologie du bassin versant. *Hydrological Sciences Bulletin*, 24(1), 37–41. doi:10.1080/02626667909491834
- Beven, K. J., Kirkby, M. J., Schofield, N., & Tagg, A. F. (1984). Testing a physically-based flood forecasting model (TOPMODEL) for three UK catchments. *Journal of Hydrology*, 69, 119–143.
- Bhunya, P. K., Berndtsson, R., Ojha, C. S. P., & Mishra, S. K. (2007). Suitability of Gamma , Chi-square , Weibull , and Beta distributions as synthetic unit hydrographs. *Journal of Hydrology*, 334(1), 28– 38. doi:10.1016/j.jhydrol.2006.09.022
- Bichot, F., Lavie, J., Dequidt, D., & Thinon-Larminach, M. (2010). Analyse des chroniques piézométriques et hydrologiques avec le logiciel TEMPO pour la gestion des prélèvements en nappe. Phase 3. RP-56481-FR.
- Bichot, F., & Pinault, J.-L. (2007). Méthodologie utilisée avec le logiciel TEMPO pour la recherche d'indicateur en nappe et la disponibilité de la ressource en eau en Poitou-Charentes.
- Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3), 268–308.
- Bonacci, O. (1993). Karst springs hydrographs as indicators of karst aquifers. *Hydrological Sciences Journal*, 38(1), 51–62.
- Boussinesq, J. (1877). *Essai sur la théorie des eaux courantes* (Vol. 2). Imprimerie nationale.

- Boussinesq, J. (1904). Recherches théoriques sur l'écoulement des nappes d'eau infiltrées dans le sol et sur le débit des sources. *Journal de Mathématiques Pures et Appliquées*, 10, 5–78.
- Charlier, J.-B. (2007). *Fonctionnement et modélisation hydrologique d'un petit bassin versant cultivé en milieu volcanique tropical. interactions.*
- Charlier, J.-B., Cattan, P., Moussa, R., & Voltz, M. (2008). Hydrological behaviour and modelling of a volcanic tropical cultivated catchment. *Hydrological Processes*, 22, 4355–4370. doi:10.1002/hyp
- Chau, K. W. (2006). Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River. *Journal of Hydrology*, 329, 363–367. doi:10.1016/j.jhydrol.2006.02.025
- Cho, J., Barone, V. A., & Mostaghimi, S. (2009). Simulation of land use impacts on groundwater levels and streamflow in a Virginia watershed. *Agricultural Water Management*, 96, 1–11. doi:10.1016/j.agwat.2008.07.005
- Cigizoglu, H., & Kisi, O. (2005). Flow prediction by three back propagation techniques using k-fold partitioning of neural network training data. *Nordic Hydrology*, 36, 49–64.
- Clausen, B. (1992). Modelling Streamflow Recession in Two Danish Streams. *Nordic Hydrology*, 23(1), 73–88.
- Conan, C., Bouraoui, F., Turpin, N., de Marsily, G., & Bidoglio, G. (2003). Modeling Flow and Nitrate Fate at Catchment Scale in Brittany (France). *Journal of Environmental Quality*, 32(6), 2026–2032.
- Coutagne, A. (1948). Etude générale des variations de débits en fonction des facteurs qui les conditionnent, 2ème partie: Les variations de débit en période non influencée par les précipitations. *La Houille Blanche*, 416–436.
- Cudennec, C., Fouad, Y., Gatot, I. S., & Duchesne, J. (2004). A geomorphological explanation of the unit hydrograph concept. *Hydrological Processes*, 18(4), 603–621. doi:10.1002/hyp.1368
- Davis, P. (1993). Levenberg-Marquardt Methods and Nonlinear Estimation. *SIAM News*, 26(6).
- Dawson, C. W., & Wilby, R. (1998). An artificial neural network approach to rainfall-runoff modelling. *Hydrological Sciences Journal*, 43(1), 47–66.
- Denić-Jukić, V., & Jukić, D. (2003). Composite transfer functions for karst aquifers. *Journal of Hydrology*, 274(1-4), 80–94. doi:10.1016/S0022-1694(02)00393-1
- Dewandel, B., Lachassagne, P., Bakalowicz, M., Weng, P., & Al-malki, A. (2003). Evaluation

- of aquifer thickness by analysing recession hydrographs . Application to the Oman ophiolite hard-rock aquifer. *Journal of Hydrology*, 274, 248–269. doi:10.1016/S0022-1694(02)00418-3
- Doherty, J. (2010). PEST: Model-Independent Parameter Estimation. User Manual: 5th Edition. Brisbane, Australia: Watermark Numerical Computing.
- Doherty, J., & Christensen, S. (2011). Use of paired simple and complex models to reduce predictive bias and quantify uncertainty. *Water Resources Research*, 47(12), 1–21. doi:10.1029/2011WR010763
- Dörfliger, N., Fleury, P., & Ladouche, B. (2009). Inverse Modeling Approach to Allogenic Karst System Characterization. *Ground Water*, 47(3), 414–426. Retrieved from <http://dx.doi.org/10.1111/j.1745-6584.2008.00517.x> DO - 10.1111/j.1745-6584.2008.00517.x
- Douez, O., & Bichot, F. (2012). *Contribution à la gestion quantitative des ressources en eau à l'aide du modèle Crétacé du sud Charentes*.
- Douez, O., Bichot, F., Dequidt, D., Dugrillon, D., Putot, E., & Petit, L. (2010). *Contribution à la gestion des prélèvements à la périphérie du Marais Poitevin par modélisation hydrodynamique*.
- Drogue, C. (1972). Analyse statistique des hydrogrammes de décrues des sources karstiques statistical analysis of hydrographs of karstic springs. *Journal of Hydrology*, 15(1), 49–68.
- Eisenlohr, L., Bouzelboudjen, M., Király, L., & Rossier, Y. (1997). Numerical versus statistical modelling of natural response of a karst hydrogeological system. *Journal of Hydrology*, 202(1-4), 244–262.
- Fernández-Martínez, J. L., & García-Gonzalo, E. (2009). The PSO family: deduction , stochastic analysis and comparison. *Swarm Intelligence*, 3(4), 245–273. doi:10.1007/s11721-009-0034-8
- Fernández-Martínez, J. L., & García-Gonzalo, E. (2012). Stochastic stability and numerical analysis of two novel algorithms of the PSO family: PP-GPSO and RR-GPSO. *International Journal on Artificial Intelligence Tools*, 21(3). doi:10.1142/S0218213012400118
- Fernández-Martínez, J. L., García-Gonzalo, E., Fernández-Muñiz, Z., & Mukerji, T. (2012). How to design a powerful family of particle swarm optimizers for inverse modelling. *Transactions of the Institute of Measurement and Control*, 34(6), 705–719. doi:10.1177/0142331211402900
- Fernández-Martínez, J. L., García-Gonzalo, E., & Naudet, V. (2010). Particle swarm optimization applied to solving and appraising the streaming-potential inverse problem. *Geophysics*, 75(4), WA3–WA15.

- Fleury, P., Plagnes, V., & Bakalowicz, M. (2007). Modelling of the functioning of karst aquifers with a reservoir model : Application to Fontaine de Vaucluse (South of France). *Journal of Hydrology*, 345, 38– 49. doi:10.1016/j.jhydrol.2007.07.014
- Frakes, B. J., & Yu, Z. (1999). An Evaluation of Two Hydrological Models for Climate Change Scenarios. *Journal of the American Water Resources Association*, 35, 1351–1363.
- Furusho, C., Thirel, G., & Andréassian, V. (2013). A simple solution for dealing with non-stationarities in urbanizing catchments with GR4J. In *Knowledge for the future 2 - IAHS - IAPSO - IASPEI Joint Assembly Gothenburg*.
- Ginn, T. R., Scheibe, T. D., Haeri, H., & McClain, C. N. (2007). Paper 40–1: An expanded survey of groundwater modeling practitioners about how they quantify uncertainty: Which tools they use, why, and why not. In *Geological Society of America Abstracts With Programs* (Vol. 39, pp. 40–41).
- Gomez, E. (2002). *Modélisation intégrée du transfert de nitrate à l'échelle régionale dans un système hydrologique. Application au bassin de la Seine*. École Nationale Supérieure des Mines de Paris.
- Griffiths, G. A., & Clausen, B. (1997). Streamflow recession in basins with multiple water storages. *Journal of Hydrology*, 190, 60–74.
- Haktanir, T., & Sezen, N. (1990). Suitability of two-parameter gamma and three-parameter beta distributions as synthetic unit hydrographs in Anatolia. *Hydrological Sciences Journal*, 35(2), 167–184.
- Hayami, S. (1951). On the propagation of flood waves. *Bulletins-Disaster Prevention Research Institute, Kyoto University*, 1, 1–16.
- Heppner, F., & Grenander, U. (1990). A stochastic nonlinear model for coordinated bird flocks. In E. Krasner (Ed.), *The ubiquity of chaos* (pp. 233–238). AAAS Publications.
- Horton, R. E. (1933a). The role of infiltration in the hydrologic cycle. *Transactions, American Geophysical Union*, 14, 446–460.
- Horton, R. E. (1933b). The role of infiltration in the hydrological cycle. *Transactions, American Geophysical Union*, 14, 446–460.
- Jain, A., Sudheer, K., & Srinivasulu, S. (2004). Identification of physical processes inherent in artificial neural network rainfall runoff models. *Hydrological Processes*, 18(3), 571–581. doi:10.1002/hyp.5502
- James, L. D., & Thompson, W. O. (1970). Least squares estimation of constants in a linear recession model. *Water Resources Research*, 6(4), 1062–1069.

- Jiang, Y., Liu, C., Huang, C., & Wu, X. (2010). Improved particle swarm algorithm for hydrological parameter optimization. *Applied Mathematics and Computation*, 217(7), 3207–3215. doi:10.1016/j.amc.2010.08.053
- Johannet, A. (2011). *Modélisation par apprentissage statistique des systèmes naturels, ou en interaction avec un environnement naturel. Applications aux karsts, crues éclair et en robotique*. Université Pierre et Marie Curie.
- Jukić, D., & Denić-Jukić, V. (2006). Nonlinear kernel functions for karst aquifers. *Journal of Hydrology*, 328, 360–374. doi:10.1016/j.jhydrol.2005.12.030
- Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. In *IEEE International Conference on Neural Networks*.
- Kennedy, J., & Spears, W. M. (1998). Matching Algorithms to Problems: An Experimental Test of the Particle Swarm and Some Genetic Algorithms on the Multimodal Problem Generator. In *Proceeding IEEE International Conference on Evolutionary Computation*.
- Koffi, Y. B., Lasm, T., Ayril, P. A., Johannet, A., Kouassi Amani, M., Assidjo, E., & Biemi, J. (2007). Optimization of Multi-Layers Perceptrons Models with Algorithms of First and Second Order. Application to the Modelling of Rainfall-Rainoff Relation in Bandama Blanc Catchment (North of Ivory Coast). *European Journal of Scientific Research*, 17(3), 313–328.
- Kong A Siou, L. (2011). *Modélisation des crues de bassins karstiques par réseaux de neurones. Cas du bassin du Lez (France)*. Université de Montpellier II.
- Kupczyk, E., Kasprzyk, A., Radczuk, L., & Czamara, W. (1994). A study of the low flow regimes of Polish rivers. In *FRIEND: Flow Regimes from International Experimental and Network Data (Proceedings of the Braunschweig Conference, October 1993)*. IAHS Publ. no. 221.
- Labat, D., Ababou, R., & Mangin, A. (1999). Linear and nonlinear input / output models for karstic springflow and flood prediction at different time scales. *Stochastic Environmental Research and Risk Assessment*, 13, 337–364.
- Lafare, A. (2011). *Modélisation mathématique de la spéléogénèse: une approche hybride à partir de réseaux de fractures discrets et de simulations hydrogéologiques*. Université Montpellier II-Sciences et Techniques du Languedoc.
- Ledoux, E. (1980). *Modélisation intégrée des écoulements de surface et des écoulements souterrains sur un bassin hydrologique*.
- Legates, D., & McCabe, G. (1999). Evaluating the use of “goodness of fit” measures in hydrologic and hydroclimatic model validation. *Water Resources Research*, 35(1), 233–241. Retrieved from <http://onlinelibrary.wiley.com/doi/10.1029/1998WR900018/full>

- Liu, S., Xu, J., Zhao, J., Xie, X., & Zhang, W. (2013). An innovative method for dynamic update of initial water table in XXT model based on neural network technique. *Applied Soft Computing Journal*, 13(10), 4185–4193. doi:10.1016/j.asoc.2013.06.024
- Luhmann, A. J., Covington, M. D., Alexander, S. C., Yi, S., Schwartz, B. F., Groten, J. T., & Alexander, E. C. (2012). Comparing conservative and nonconservative tracers in karst and using them to estimate flow path geometry. *Journal of Hydrology*, 448-449, 201–211. doi:10.1016/j.jhydrol.2012.04.044
- MacDonald, M. G., & Harbaugh, A. W. (1988). A modular three-dimensional finite difference groundwater flow model. *Modelling Techniques, Techniques of Water Resources Investigation of the USGS*.
- Maillet, E. T. (1905). *Essais d'hydraulique souterraine & fluviale*. Hermann.
- Mangin, A. (1975). *Contribution à l'étude hydrodynamique des aquifères karstiques*. Université de Dijon.
- Moore, R. D. (1997). Storage-outflow modelling of streamflow recessions , with application to a shallow-soil forested catchment. *Journal of Hydrology*, 198, 260–270.
- Moussa, R. (1996). Analytical Hayami solution for the diffusive wave flood routing problem with lateral inflow. *Hydrological Processes*, 10(9), 1209–1227.
- Moussa, R., & Bocquillon, C. (1996). Algorithms for solving the diffusive wave flood routing equation. *Hydrological Processes*, 10(1), 105–123.
- Moussu, F. (2011). *Prise en compte du fonctionnement hydrodynamique dans la modélisation pluie débit des systèmes karstiques*. Université Pierre et Marie Curie-Paris VI.
- Nash, J. E. (1957). The form of the instantaneous unit hydrograph. *IAHS Publ*, 45(3), 114–121.
- Nash, J. E., & Sutcliffe, J. V. (1970). River flow forecasting through conceptual models part I—A discussion of principles. *Journal of Hydrology*, 10(3), 282–290.
- Padilla, A., Pulido-Bosch, A., & Mangin, A. (1994). Relative importance of baseflow and quickflow from hydrographs of karst spring. *Ground Water*, 32(2), 267–277.
- Paquet, E. (2004). Évolution du modèle hydrologique MORDOR : modélisation du stock nival à différentes altitudes. *La Houille Blanche*, 75–82.
- Pauwels, H., Petelet, E., & Bichot, F. (2008). CAMERA CAractérisation et Modélisation de l'Evolution des tRAnsferents des pollutions diffuses – Application aux pollutions d'origine agricole et urbaine dans un contexte d'aquifères calcaires multicouches faillés Bassin de la Sèvre-Niortaise à l'amont de N.

- Perrin, C., Michel, C., & Andréassian, V. (2003). Improvement of a parsimonious model for streamflow simulation. *Journal of Hydrology*, 279, 275–289. doi:10.1016/S0022-1694(03)00225-7
- Pinault, J.-L. (2007). Manuel utilisateur de TEMPO Logiciel de traitement et de modélisation des séries temporelles en hydrogéologie et en hydrogéochimie. BRGM/RP-55313-FR.
- Pinault, J.-L., Amraoui, N., & Golaz, C. (2005). Groundwater-induced flooding in macropore-dominated hydrological system in the context of climate changes. *Water Resources Research*, 41. doi:10.1029/2004WR003169
- Pinault, J.-L., & Berthier, F. (2007). A methodological approach to characterize the resilience of aquatic ecosystems with application to Lake Annecy, France. *Water Resources Research*, 43. doi:10.1029/2006WR005125
- Pinault, J.-L., Pauwels, H., & Cann, C. (2001). Inverse modeling of the hydrological and the hydrochemical behavior of hydrosystems: Application to nitrate transport and denitrification. *Water Resources Research*, 37(8), 2179–2190.
- Pinault, J.-L., Plagnes, V., Aquilina, L., & Bakalowicz, M. (2001). Inverse modeling of the hydrological and the hydrochemical behavior of hydrosystems : Characterization of karst system functioning. *Water Resources Research*, 37(8), 2191–2204.
- Pinault, J.-L., & Schomburgk, S. (2006). Inverse modeling for characterizing surface water/groundwater exchanges. *Water Resources Research*, 42. doi:10.1029/2005WR004587
- Pinder, G. F., & Jones, J. F. (1969). Determination of the ground-water component of peak discharge from the chemistry of total runoff. *Water Resources Research*, 5(2), 438–445.
- Pistre, S., Hakoun, V., & Falgayettes, P. (2013). Cartographie des drains karstiques noyés par une sonde autonome semi-flottante. In *19èmes journées techniques du Comité Français d'Hydrogéologie de l'Association Internationale des Hydrogéologues*. (pp. 135–136).
- Plantier Bornancin, A. (2013). *Conception de modèles de prévision des crues éclair par apprentissage artificiel*. Université Pierre et Marie Curie.
- Poeter, E. P., & Hill, M. C. (1997). Inverse Models: A Necessary Next Step in Ground-Water Modeling. *Ground Water*, 35(2), 250–260. Retrieved from <http://dx.doi.org/10.1111/j.1745-6584.1997.tb00082.x> DO - 10.1111/j.1745-6584.1997.tb00082.x
- Refsgaard, J. C., & Storm, B. (1996). Construction, calibration and validation of hydrological models. In *Distributed hydrological modelling* (pp. 41–54). Springer.
- Robinson, J., & Rahmat-Samii, Y. (2004). Particle Swarm Optimization in Electromagnetics. *Antennas and Propagation, IEEE Transactions*, 52(2), 397–407.

- Samani, N., & Ebrahimi, B. (1996). Analysis of spring hydrographs for hydrogeological evaluation of a karst aquifer system. *Theoretical & Applied Karstology*, 9, 97–112.
- Shamseldin, A. Y. (1997). Application of a neural network technique to rainfall-runoff modelling. *Journal of Hydrology*, 199(3), 272–294.
- Sherman, L. K. (1932). Streamflow from rainfall by the unit-graph method. *Engineering News Record*, 108, 501–505.
- Singh, P. K., Bhunya, P. K., Mishra, S. K., & Chaube, U. C. (2007). An extended hybrid model for synthetic unit hydrograph derivation. *Journal of Geophysical Research*, 334, 347–360. doi:10.1016/j.jhydrol.2007.01.006
- Singh, V. P. (1988). *Hydrologic systems. Rainfall-runoff modeling. Volume 1*. Prentice Hall.
- Tada, T. (2007). Parameter Optimization of Hydrological Model Using the PSO Algorithm. *Journal of Japan Society of Hydrology and Water Resources*, 20(5), 450–461.
- Tallaksen, L. M. (1995). A review of baseflow recession analysis. *Journal of Hydrology*, 165, 349–370.
- Tarantola, A. (2005). *Inverse problem theory and methods for model parameter estimation*. SIAM.
- Thiéry, D. (1988). Forecast of changes in piezometric levels by a lumped hydrological models. *Journal of Hydrology*, 97, 129–148.
- Thiéry, D. (1993). Calage automatique des modeles hydrodynamiques maillés. Détermination des zones géographiques homogenes et des parametres optimaux associés. Application à 5 systemes aquiferes. *Rev. Hydrogéol*, 4, 281–291.
- Thiéry, D. (1994). Calibration of groundwater models by optimization of parameters in homogeneous geological zones. In *Stochastic and Statistical Methods in Hydrology and Environmental Engineering* (pp. 69–82). Springer.
- Thiéry, D. (2003). *Logiciel GARDÉNIA version 6.0*.
- Thiéry, D. (2004). *Discrétisation et équations résolues dans le code MARTHE*.
- Thiéry, D. (2006). *Didacticiel du processeur WinMarthe v4.0*. Retrieved from <http://infoterre.brgm.fr/rapports/RP-54652-FR.pdf>
- Thiéry, D. (2012a). *Code MARTHE 7.4 : Couplage Nappe-Rivières et Nappe-Drains Bilans Hydroclimatiques*.
- Thiéry, D. (2012b). *Échanges Nappe Rivière dans le code de calcul MARTHE*.

- Thiéry, D., & Moutzopoulos, C. (1992). Un modèle hydrologique spatialisé pour la simulation des très grands bassins : le modèle EROS, formé de grappes de modèles globaux élémentaires. In *VIIIe journées hydrologiques - Orstom - Septembre 1992* (pp. 285 – 295).
- Toebes, C., & Strang, D. (1964). On recession curves. *Journal of Hydrology (New Zealand)*, 3(2), 2–14.
- Tokar, A. S., & Johnson, P. A. (1999). Rainfall-runoff modeling using artificial neural networks. *Journal of Hydrologic Engineering*, 4(3), 232–239.
- Tokar, A. S., & Markus, M. (2000). Precipitation-runoff modeling using artificial neural networks and conceptual models. *Journal of Hydrologic Engineering*, 5(2), 156–161.
- Viville, D., Ladouche, B., & Bariac, T. (2006). Isotope hydrological study of mean transit time in the granitic Strengbach catchment (Vosges massif, France): application of the FlowPC model with modified input function. *Hydrological Processes*, 20(8), 1737–1751. Retrieved from <http://dx.doi.org/10.1002/hyp.5950> DO - 10.1002/hyp.5950
- von Asmuth, J. R., Bierkens, M. F. P., & Maas, K. (2002). Transfer function-noise modeling in continuous time using predefined impulse response functions. *Water Resources Research*, 38(12). doi:10.1029/2001WR001136
- von Asmuth, J. R., Maas, K., Knotters, M., Bierkens, M. F. P., Bakker, M., Olsthoorn, T. N., ... Asmuth, D. C. Von. (2012). Software for hydrogeologic time series analysis , interfacing data with physical insight. *Environmental Modelling & Software*, 38, 178–190. doi:10.1016/j.envsoft.2012.06.003
- Weiler, M., Mcglynn, B. L., Mcguire, K. J., & McDonnell, J. J. (2003). How does rainfall become runoff? A combined tracer and runoff transfer function approach. *Water Resources Research*, 39(11), 1–13. doi:10.1029/2003WR002331
- Wittenberg, H. (1999). Baseflow recession and recharge as nonlinear storage processes. *Hydrological Processes*, 13, 715–726.
- Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67–82.
- Yang, X.-S. (2012). Swarm-Based Metaheuristic Algorithms and No-Free-Lunch Theorems. In R. Parpinelli (Ed.), *Theory and New Applications of Swarm Intelligence*. InTech.

Modélisation des hydrosystèmes par approche systémique

Résumé

Dans l'état actuel des connaissances, il est impossible de poser correctement toute la physique permettant de modéliser les hydrosystèmes dans leur ensemble, notamment à cause de la dynamique très contrastée des différents compartiments. Les modèles systémiques simplifient la représentation des hydrosystèmes en ne considérant que leurs flux d'échange. L'objet de ce travail est de proposer un outil de modélisation systémique fournissant des informations sur le fonctionnement physique des hydrosystèmes, tout en étant simple et parcimonieux. Ce modèle nommé MASH (pour Modélisation des Hydrosystèmes par Approche Systémique) est basé sur l'utilisation de fonctions de transfert paramétriques choisies en fonction de leur faible paramétrisation, leur caractère général et leur interprétation physique. Il est versatile, dans le sens que son architecture est modulable et que le nombre d'entrées, le nombre de fonctions de transfert en série et le type de fonctions de transfert utilisé est laissée à la discrétion de l'utilisateur. Ce modèle est inversé en utilisant de récentes avancées en apprentissage automatique grâce à une famille d'heuristiques basée sur l'intelligence en essaim nommé « optimisation par essaim de particule » (ou PSO pour « *Particle Swarm Optimization* »). Le modèle et ses algorithmes d'inversion sont testés sur un cas d'école synthétique, puis sur un cas d'application réel.

Mots clés : Apprentissage automatique, Optimisation par essaim de particule, Fonction de transfert, Approche systémique

Hydrosystem Modelling with a Systemic Approach

In the light of current knowledge, hydrosystems cannot be modelled as a whole since underlying physical principles are not totally understood. Systemic models simplify hydrosystem representation by considering only water flows. The aim of this work is to provide a systemic modelling tool giving information about hydrosystem physical behavior while being simple and parsimonious. This model, called HMSA (for Hydrosystem Modelling with a Systemic Approach) is based on parametric transfer functions chose for their low parametrization, their general nature and their physical interpretation. It is versatile, since its architecture is modular, and the user can choose the number of inputs, outputs and transfer functions. Inversion is done with recent machine learning heuristic family, based on swarm intelligence called PSO (Particle Swarm Optimization). The model and its inversion algorithms are tested first with a textbook case, and then with a real-world case.

Keywords: Machine learning, Particle swarm optimization, Transfer function, Systemic approach