



HAL
open science

Propagation d'incertitudes par analyse de sensibilité dans un modèle numérique avec prise en compte du calage

J.B. Faure

► **To cite this version:**

J.B. Faure. Propagation d'incertitudes par analyse de sensibilité dans un modèle numérique avec prise en compte du calage. SimHydro 2010, Jun 2010, Sophia-Antipolis, France. 8 p. hal-00558772

HAL Id: hal-00558772

<https://hal.science/hal-00558772>

Submitted on 24 Jan 2011

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.

PROPAGATION D'INCERTITUDES PAR ANALYSE DE SENSIBILITÉ DANS UN MODÈLE NUMÉRIQUE AVEC PRISE EN COMPTE DU CALAGE

*Uncertainty propagation using sensitivity analysis in a numerical
model taking calibration into account*

Jean-Baptiste FAURE

Unité de recherche Hydrologie-Hydraulique, Cemagref,
3bis quai Chauveau 69336 Lyon Cedex 09 – FRANCE

jean-baptiste.faure@cemagref.fr

MOTS CLÉS

Hydraulique fluviale, équations de Saint-Venant.
Fluvial hydraulics, shallow water equations.

RÉSUMÉ

Cette communication présente une méthode d'évaluation de l'incertitude du résultat d'un modèle numérique en fonction de l'incertitude estimée sur ses paramètres d'entrée. Ici « modèle numérique » désigne l'ensemble code de calcul et jeu de données. Pour ce travail nous voulions nous affranchir de l'usage des probabilités qui sont mal comprises par bien des utilisateurs de modèles numériques et des destinataires de leurs résultats. Nous avons donc préféré aborder la question par le biais de l'analyse de sensibilité en évaluant l'incertitude sur la réponse du modèle numérique par une mesure de l'enveloppe des réponses possibles.

ABSTRACT

This paper presents a method for evaluating the uncertainty of the result of a numerical model based on the uncertainty estimated with respect to its input parameters. Here, "numerical model" means the set of computer code and data set. For this work we wanted do not use probabilities that are poorly understood by many users of numerical models and recipients of their results. We therefore preferred to address the issue through sensitivity analysis by evaluating the uncertainty on the response of the numerical model by measuring the envelope of possible computed results.

1. INTRODUCTION

Notre approche de l'évaluation des incertitudes se distingue des approches probabilistes habituelles en ce sens que nous cherchons plutôt à encadrer toutes les valeurs possibles de la grandeur évaluée par le modèle – ici la surface inondée – compte tenu des incertitudes sur les données, estimées a priori. L'idée qui sous-tend ce choix est que, pour l'utilisateur final, définir une incertitude en termes de probabilités est difficile à interpréter et par conséquent de peu d'utilité d'un point de vue opérationnel c'est à dire quand il s'agit de prendre des décisions. Par exemple quelle est l'utilité opérationnelle de dire que la probabilité que la limite de la zone inondée se trouve entre ces deux lignes est de 90% ? Cette probabilité peut signifier que si on réalise 1000 fois la même simulation avec des coefficients de Strickler différents, à peu près 900 résultats tomberont entre ces deux lignes. Cependant il est important d'un point de vue opérationnel, par exemple en période de crise, d'avoir une idée de ce qui se passe pour les 100 résultats qui tombent en dehors. Si on soupçonne qu'ils pourraient tomber loin des limites à 90% il serait sans doute raisonnable et prudent de déterminer l'incertitude à 95% voire à 99%. De même quand on voit la difficulté de beaucoup à comprendre ce que signifie « crue centennale », il est clair que l'utilisation de probabilités doit être faite avec beaucoup de prudence afin

de ne pas ouvrir la porte à des interprétations erronées susceptibles d'augmenter la confusion à propos de la précision des modèles numériques et de la confiance qu'on peut mettre dans leurs résultats.

Une autre approche [1, 2] de l'évaluation des incertitudes que nous avons étudiée, est basée sur le calcul du gradient de la solution (ici la surface inondée) en fonction des paramètres d'entrée considérés comme incertains. Le code de calcul est considéré comme une fonction mathématique qui calcule la surface inondée en fonction d'une variable multidimensionnelle constituée par les coefficients de Stricklers. Pour calculer son gradient on peut procéder par approximation en calculant les taux de variation pour chaque direction de l'espace des paramètres. On peut aussi procéder de façon analytique en considérant le modèle comme la composition (au sens mathématique) de plusieurs fonctions élémentaires (en fait chaque ligne de code), il « suffit » alors de produire un code dérivé qui réalise la dérivation de cette fonction composée. Le nombre de lignes de code se comptant en dizaines de milliers ce serait un travail très lourd si on voulait le faire à la main. On utilise de préférence un logiciel de « différenciation automatique » qui permet de le faire de façon *presque* automatique.

Cette approche présente plusieurs difficultés :

- Difficultés d'ordre pratique : la différenciation d'un code existant qui n'a pas été conçu dès le départ dans ce but est, en fait très compliquée et seulement partiellement automatique. Un code hydraulique utilisable pour traiter des problèmes réels ne se limite pas à la résolution des équations de Saint-Venant, il prend aussi en compte des singularités hydrauliques et des modèles complémentaires qui introduisent des difficultés particulières comme des discontinuités dans les lois d'ouvrages ou dans les différents régimes d'écoulement (seuils de débordement). En conséquence tout changement dans le code d'origine ou dans le choix des paramètres d'entrée ou sur la variable de sortie, impose un gros travail sur le code source avec les conséquences qu'on imagine facilement quand l'objectif est de produire un logiciel opérationnel et publiable.
- Difficulté théorique : le gradient est une bonne mesure de la sensibilité d'une fonction aux variations de ses paramètres d'entrée, mais c'est une mesure locale. Or dans notre cas nous voulons évaluer la sensibilité de notre modèle à des variations de stricklers qui n'ont rien de locales. En général l'incertitude estimée sur les stricklers est plutôt de l'ordre de plusieurs unités ce qui n'est pas petit. Il faudrait donc calculer le gradient dans tout l'espace des paramètres et en étudier les variations. Cela supposerait donc, en particulier, de travailler sur les dérivées secondes du modèle alors que les dérivées premières sont déjà compliquées à obtenir. De plus le calcul des dérivées premières et secondes n'est pas suffisant à lui seul pour évaluer les incertitudes. La question est donc de savoir, quand bien même on aurait un moyen pratique d'évaluer la sensibilité locale du modèle, comment on utilise ce moyen pour passer à une évaluation plus globale de l'incertitude. On peut certes utiliser ce moyen pour mettre en œuvre un algorithme d'optimisation afin de calculer les surfaces inondées minimales et maximales mais la complexité du logiciel à produire ne plaide pas en faveur de cette approche alors qu'on peut résoudre le même problème d'optimisation par des moyens nettement moins coûteux, au moins en termes de développement logiciel.

2. PRÉSENTATION DE LA MÉTHODE

2.1. Mise en forme du problème

L'incertitude est évaluée par l'écart entre les surfaces inondées maximale et minimale relativement à la surface inondée obtenue pour les valeurs médianes des stricklers. On peut calculer cet écart en résolvant deux problèmes d'optimisation distincts, l'un pour la surface minimale et l'autre pour la surface maximale. On peut aussi le calculer en résolvant un seul problème d'optimisation consistant à maximiser l'écart entre deux surfaces inondées. Bien entendu ce problème d'optimisation prend en compte deux fois plus de variables et contraintes. On a pu vérifier sur les exemples traités que les deux approches donnent les mêmes résultats mais la seconde est en fait moins coûteuse.

Précisément on considère le modèle comme une fonction mathématique qui reçoit en entrée les coefficients de Strickler pour fournir en sortie la surface inondée. Cette surface inondée est calculée pour la ligne d'eau enveloppe formées des niveaux maximaux atteints en chaque point de calcul. Pour chaque point de calcul on obtient à partir du niveau maximal une largeur du plan d'eau ; la surface inondée dont il est question est l'intégrale de ces largeurs le long du tronçon de rivière. Si k est le vecteur des stricklers, $M(k)$ est donc la surface inondée pour l'événement hydrologique considéré (la crue). On appelle k_0 le meilleur jeu de stricklers obtenu par calage et on cherche à évaluer l'incertitude portant sur $M(k)$ quand k parcourt le domaine admissible déterminé par le modélisateur. Par commodité on suppose que ce domaine K est un produit cartésien d'intervalles centrés en k_0 .

Le problème d'optimisation consiste donc à trouver deux jeux de stricklers qui réalisent le maximum de :

$$\frac{|M(k_1) - M(k_2)|}{M(k_0)} \quad (1)$$

avec les contraintes :

$$k_i^{\min} \leq k_i^j \leq k_i^{\max} \quad \forall i=1,2 \text{ et } j=1,\dots,2n \quad (2)$$

où n est le nombre de secteurs de résistance à l'écoulement homogènes (où les stricklers sont donc constants). À ces contraintes de base il faut ajouter celles qui concernent le calage du modèle :

$$C(k) \leq \epsilon \text{ pour } k \in K \quad (3)$$

où $C(k)$ mesure l'écart entre les observations et le calcul pour l'événement de calage et ϵ est un seuil, choisi pas le modélisateur, qui représente l'erreur de calage acceptable. Nous avons choisi l'écart moyen :

$$C(k) = \frac{1}{m} \sum_{i=1}^m |Z(x_i) - Z_{obs}(x_i)| \quad (4)$$

où m est le nombre d'observations (laisses de crue), les x_i sont les points d'observation, $Z(x)$ la cote maximale calculée au point x pour l'événement de calage et $Z_{obs}(x)$ la cote maximale observée au point x . Le problème d'optimisation s'exprime alors de la façon suivante :

Trouver k_1 et k_2 dans K tels que :

$$\frac{|M(k_1) - M(k_2)|}{M(k_0)} \geq \frac{|M(k) - M(k')|}{M(k_0)} \quad \forall k \in K \text{ et } k' \in K \quad (5)$$

sous la contrainte $C(k) \leq \epsilon$ et $C(k') \leq \epsilon$

2.2. Méthode de résolution.

M étant un code de calcul résolvant un système d'équations non linéaires, nous n'avons pas d'information sur la forme de la surface $M(k)$. En particulier nous ne savons pas si elle a des minimums et maximums globaux ou seulement des extremums locaux. Nous ne savons pas non plus si la fonction M est dérivable partout dans K . Nous avons donc opté pour une méthode d'optimisation globale qui ne fait aucun hypothèse particulière sur la fonction objectif et en particulier ne requiert pas le calcul de son gradient et donc pas non plus celui de M . Notre choix s'est porté sur l'algorithme du Recuit-Simulé [3]. Des essais exploratoires avec un algorithme génétique (*Pikaia*) ont été réalisés mais, devant la pauvreté des résultats obtenus, n'ont pas été poursuivis .

L'idée de base de l'algorithme du Recuit-Simulé est de faire des modifications aléatoires des consignes (ici les stricklers) et de conserver celles qui améliorent la solution estimée mais aussi des consignes qui ne l'améliorent pas en les choisissant selon un critère probabiliste dépendant d'un paramètre appelé température. Plus cette température est élevée plus une consigne qui n'apporte pas d'amélioration a une probabilité élevée d'être conservée malgré tout. La température est un paramètre destiné à décroître au cours des itérations de sorte qu'au début on conserve beaucoup de consignes

alors que vers la fin on ne conserve plus que des consignes qui améliorent vraiment la solution estimée. Cette méthode permet d'empêcher l'algorithme de rester bloqué sur un extremum local. Si on se représente la fonction objectif comme une surface avec des creux et de bosses et en supposant qu'on cherche le creux le plus profond, l'algorithme permet de faire sortir la solution courante d'un puits pour peu que la température soit suffisante.

L'algorithme du Recuit-Simulé comporte plusieurs étapes cruciales pour son efficacité, la méthode de modification aléatoire des consignes, la stratégie de réduction de la température, la prise en compte des contraintes et le test d'arrêt. Il se compose en fait de deux boucles itératives imbriquées, la première, extérieure, itère sur la température et la seconde, intérieure et à température fixée, recherche, par modification aléatoire des consignes, une meilleure estimation de la solution. Il y a donc deux tests d'arrêt, un pour chaque boucle. La boucle sur la température est poursuivie tant que la température n'a pas atteint une valeur plancher fixée à l'avance. On peut donc prévoir le nombre de niveaux de température qui seront parcourus. Ce nombre dépend de la façon dont on fait décroître la température d'une itération à la suivante. On peut imaginer une décroissance linéaire ($T_{n+1} = T_n - dT$) ou exponentielle ($T_{n+1} = \alpha \cdot T_n$) ou encore une courbe de décroissance en S comme nous l'avons finalement fait.

Le test d'arrêt pour la boucle intérieure est moins simple. Nous n'avons pas trouvé mieux que d'arrêter les itérations quand il ne passe plus grand-chose. Pour cela on compte les itérations successives qui n'apportent aucune amélioration de l'optimum courant et on interrompt les itérations si leur nombre dépasse un certain plafond fixé a priori.

2.3. Description détaillée de la méthode

2.3.1. Algorithme du Recuit-Simulé (maximisation)

On cherche w_{opt} dans $[w_{min}; w_{max}]$ qui maximise la valeur de $F(w)$.

Initialisation :

$n = 0$ (compteur d'itérations) ; $T = T_0$ (Température initiale / maximale)

solution initiale : $w = 0,5 \cdot (w_{max} - w_{min})$

calculer $F(w)$: estimation initiale ; $w_{opt} = w_{actuel} = w$

Répéter tant que $T > T_{min}$

$w_{actuel} = w_{opt}$; critère_arrêt = faux ; $m = 0$: compteur des améliorations de w_{opt}

Répéter tant que critère_arrêt = faux

Générer aléatoirement un nouvelle consigne w

Si $F(w_{opt}) < F(w)$ alors $w_{opt} = w$ et $w_{actuel} = w_{opt}$ et $m = 0$

Sinon si $F(w_{actuel}) < F(w)$ alors $w_{actuel} = w$ et $m = m+1$

Sinon si $random[0,1] < \exp\left(-\frac{F(w_{actuel}) - F(w_n)}{T}\right)$ alors $w_{actuel} = w$

Sinon $m = m+1$

critère_arrêt = $(m > m_{max})$! m_{max} évaluations sans amélioration de l'optimum.

Fin répétition

Changer la température : $T = g(T,n)$; $n = n+1$

Fin répétition

Solution = w_{opt}

2.3.2. Calcul de la consigne

En notation vectorielle la consigne est obtenue par :

$$w = random[w_{actuel} - \delta w ; w_{actuel} + \delta w] \text{ avec } \delta w = \alpha \cdot (w_{max} - w_{min}) \quad (6)$$

Le facteur d'échelle α peut être modifié en fonction de la température et d'autre critères. Par exemple on peut modifier sa valeur à la baisse ou à la hausse pour s'assurer que le taux de consignes à évaluer ne satisfaisant pas les contraintes de calage reste suffisamment petit. D'autre part un facteur d'échelle trop petit réduit l'éventail des possibilités pour les valeurs de consignes ; il faut donc pouvoir l'augmenter. Dans le cas où l'une des composantes de w dépasse les frontières du domaine, cette

composante est ramenée sur la frontière. Cette correction permet de trouver plus facilement la solution quand elle touche la frontière du domaine.

2.3.3. La fonction objectif

La fonction objectif dépend en fait d'une double consigne : $w = (w_1, w_2)$ où w_i est un vecteur de coefficients de Strickler. Elle prend en compte la contrainte sur le calage sous la forme d'une fonction de pénalisation :

$$F(w_1, w_2) = 100 \cdot \frac{|S(w_1) - S(w_2)|}{S(w_0)} + \lambda \cdot (\max(0, Ec(w_1) - \epsilon) + \max(0, Ec(w_2) - \epsilon)) \quad (7)$$

où $Ec(w)$ mesure l'écart entre les niveaux observés et les résultats du calcul. La valeur ϵ permet de prendre en compte une incertitude estimée sur les observations de calage.

2.3.4. Décroissance de la température

La fonction de décroissance de la température est construite pour avoir une décroissance lente au début afin de permettre une exploration complète du domaine de recherche de l'optimum (recherche globale). Ensuite la décroissance est assez rapide puis redevient lente pour affiner l'estimation de l'optimum par une recherche locale. La fonction est définie comme le maximum des courbes bleue et rouge sur la figure 1. Une autre fonction de température a été testée en alliant une décroissance seulement linéaire au début pour terminer par une décroissance exponentielle pour les températures basses donc une décroissance plus lente.

2.3.5. Contrôle de l'algorithme

Plusieurs paramètres permettent de contrôler l'algorithme du Recuit-Simulé tel qu'utilisé ici :

- la température initiale ;
- la décroissance de la température (exponentielle, linéaire, mixte, etc.) ;
- le critère utilisé pour changer de température (nombre d'évaluations fixe ou nombre d'évaluation maximal sans amélioration de l'optimum) ;
- la taille du domaine dans lequel on génère aléatoirement les nouvelles consignes ; cette taille peut être fixe ou décroître avec la température ou bien encore être ajustée en fonction d'un autre critère.

Les nombreuses expériences numériques réalisées suggèrent qu'il n'existe pas de stratégie qui garantisse à tous les coups la convergence la plus rapide. Une chose est claire c'est qu'il faut beaucoup d'évaluations de la fonction objectif pour obtenir une bonne estimation de l'optimum.

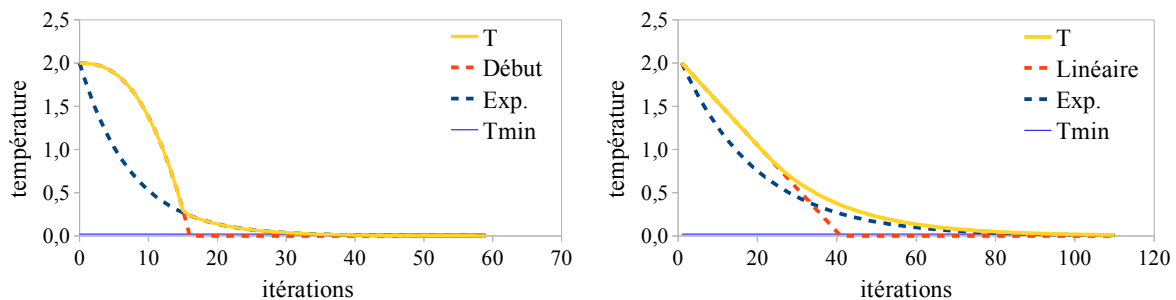


figure 1 : fonctions de contrôle de la température

3. EXPÉRIENCES NUMÉRIQUES

Le cas testé est un tronçon de 7 km de la rivière Saar [1, 4] dont les données sont disponibles sur demande. Les événements hydrologiques utilisés sont fictifs mais représentatifs de ce qui peut se passer sur cette rivière. La crue de calage (figure 2) est définie par un débit de pointe de 500 m³/s alors que la crue sur laquelle porte l'analyse de sensibilité a un débit de pointe de 900 m³/s. Cet écart est

assez représentatif de la pratique courante des études hydrauliques car on est bien obligé de se contenter des données disponibles.

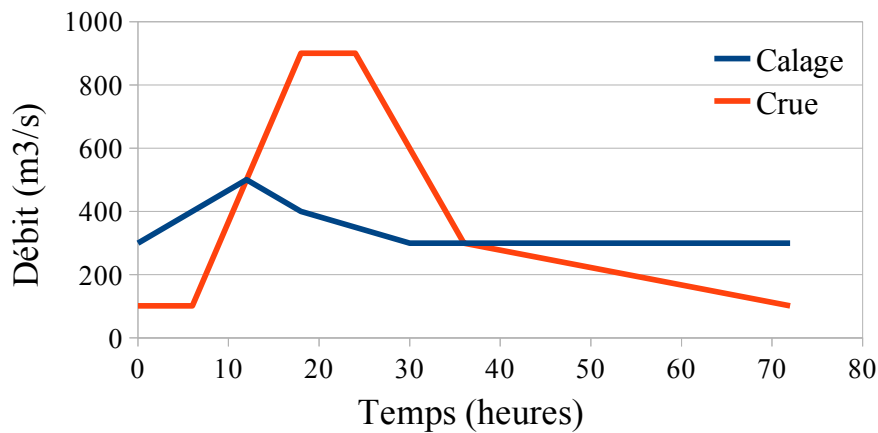


figure 2 : hydrogrammes de crue

Pour la crue de calage nous disposons de 3 points d'observation où nous avons chaque fois un niveau maximal atteint. Il s'agit bien sûr d'une crue observée simulée avec des coefficients de Strickler choisis, voisins de l'estimation a priori de 25 pour le lit mineur et 15 pour le lit majeur actif. Nous avons évalué la sensibilité du modèle pour 4 situations différentes, selon le nombre de tronçons homogènes choisis. Nos tronçons ont des longueurs à peu près identiques. Nous avons comparé les cas à 1, 3, 6 et 14 tronçons. Dans tous les cas les stricklers sont recherchés dans l'intervalle [21 ; 29] pour le lit mineur et [10 ; 20] pour le lit majeur actif et le critère d'erreur sur le calage est de 10 cm. Le code de calcul utilisé pour réaliser les simulations hydrauliques est le code MAGE développé par l'auteur [5, 6]. Nous avons évalué la sensibilité du modèle dans les 4 cas pour l'événement de calage et pour la crue étudiée. L'écart est mesuré par rapport à la surface inondée obtenue pour les stricklers moyens c'est à dire ici 25 et 15. Les résultats indiqués sont bien sûr des valeurs a priori sous-estimées. Dans tous les cas il faut plusieurs milliers de simulations pour obtenir une bonne convergence [4].

	Crue de calage (500 m ³ /s)	Crue étudiée (900 m ³ /s)
Surface inondée moyenne	15,06 ha	51,09 ha
1 tronçon	> 30,13%	> 27,76%
3 tronçons	> 35,66%	> 31,31%
6 tronçons	> 52,11%	> 47,89%
14 tronçons	> 57,65%	> 51,40%

Tableau 1 : résultats de l'optimisation

3.1. Analyse des résultats

Les résultats numériques présentés ici n'ont pas d'intérêts pour eux mêmes mais seulement pour ce qu'ils révèlent de la méthode d'estimation de la sensibilité du modèle numérique. Une première remarque est que même pour la crue de calage un modèle numérique peut être très sensible aux coefficients de Strickler ; pourtant tous ces jeux de coefficients sont équivalents si on ne considère que la reproduction des laisses de crue. On peut donc en déduire que ce critère peut être insuffisant pour caler correctement un modèle destiné à déterminer des zones inondables.

Une deuxième remarque est que l'incertitude augmente quand on augmente le nombre de tronçons c'est à dire quand on augmente le nombre de paramètres. C'est assez prévisible car on a ainsi plus de coefficients de Strickler qui ne sont pas sous l'influence hydraulique directe des données de calage. Ainsi il apparait que certains coefficients de Strickler n'ont aucune influence sur la qualité du calage sur les niveaux, mais font la différence pour la surface inondée. Cela est du en partie au fait que la crue utilisée pour le calage est nettement plus faible que la crue dont on teste la surface inondée, les coefficients de Strickler du lit majeur sont donc moins déterminants pour le calage. Cela révèle que

notre modèle calé n'est pas suffisamment prédictif quand on veut l'utiliser pour des crues sensiblement plus fortes que la crue de calage. On a donc ici un moyen d'invalider un modèle calé si on considère que sa sensibilité aux paramètres de calage ne devrait pas être beaucoup plus forte pour les crues étudiées qu'il ne l'est pour la ou les crues de calage. Hors ici, même si en valeurs relatives les sensibilités sont comparables, en valeurs absolues les écarts pour la crue étudiée sont bien plus grand que pour la crue de calage.

3.2. Comparaison avec la méthode de Monte-Carlo

On obtient de meilleurs résultats (tableau 2) avec la méthode d'optimisation qu'avec une méthode d'exploration de type Monte-Carlo consistant à réaliser autant de simulations choisies aléatoirement pour qu'un nombre suffisant satisfassent le critère de calage. Pour un nombre de simulations efficaces (celles qui satisfont les contraintes de calage) équivalent, cette approche se révèle sensiblement moins précise quand le nombre de variables à prendre en compte dans l'optimisation augmente. Les essais sont conduits en générant, comme pour l'optimisation, des consignes qui peuvent être situées hors du domaine et en appliquant à ces consignes une troncature pour les ramener à la frontière du domaine la plus proche. On a ainsi des chances d'obtenir des consignes sur les bords du domaine. Le taux de consignes extérieures est de l'ordre de 1%. Sur le cas à 14 tronçons on obtient exactement les mêmes résultats avec un taux de 5%. On remarque sur ces résultats que les estimations obtenues par la méthode de Monte-Carlo sont dans tous les cas nettement moins bonnes que celles obtenues par optimisation. De plus le nombre réel de simulations à réaliser est bien plus grand que pour l'optimisation pour laquelle on obtient une bonne convergence avec environ 10000 simulations effectives.

	3000 simulations	6000 simulations	9000 simulations
Crue 900 m³/s et 14 tronçons : Taux de simulations efficaces = 64%			
Surface maximale en ha	61,62	63,12	63,12
Surface minimale en ha	46,45	46,45	46,45
Écart en % / surface moyenne	29,69	32,63	32,63
Crue 900 m³/s et 6 tronçons Taux de simulations efficaces = 41%			
Surface maximale en ha	64,31	64,31	64,31
Surface minimale en ha	45,98	45,98	45,77
Écart en % / surface moyenne	35,88	35,88	36,29
Crue 900 m³/s et 3 tronçons Taux de simulations efficaces = 15%			
Surface maximale en ha	59,80	59,80	59,80
Surface minimale en ha	46,45	46,45	46,45
Écart en % / surface moyenne	26,13	26,13	26,13
Crue 900 m³/s et 1 tronçon Taux de simulations efficaces = 28%			
Surface maximale en ha	60,34	60,60	60,65
Surface minimale en ha	46,59	46,54	46,53
Écart en % / surface moyenne	26,91	27,52	27,64

Tableau 2 : résultats de la méthode de Monte-Carlo

4. CONCLUSION

L'introduction, par rapport à des travaux précédents [4], d'une contrainte sur le calage permet d'affiner l'estimation de la sensibilité d'un modèle numérique par rapport aux coefficients de Strickler. Elle

montre que la tolérance sur le calage, si elle donne des lignes d'eau enveloppe équivalentes pour l'événement de calage, laisse une grande marge d'incertitude sur la surface inondée. Ainsi une marge d'erreur de 0,1 m sur la reproduction de niveaux des laisses de crue pour l'événement de calage permet, sur le cas très simple étudié, une variation de l'ordre de 30 à 50% par rapport à la surface inondée obtenue avec le jeu de coefficients de Strickler moyens. Pourtant tous ces jeux de coefficients sont équivalents du point de vue du calage. Des simulations analogues, quoique plus coûteuses, pourraient facilement être réalisées pour étudier la sensibilité du même modèle par rapport à des données géométriques pour tenir compte, par exemple, de l'incertitude des mesures d'altitude.

5. RÉFÉRENCES

- [1] Souhar O., Faure J-B., (2010) Automatic differentiation strategy for the local sensitivity analysis of a one-dimensional hydraulic model, *International Journal for Numerical Methods in Fluids*, (à paraître).
- [2] Souhar O., Faure J-B., (2009) Approach for uncertainty propagation and design in Saint Venant equations via automatic sensitive derivatives applied to Saar river, *Revue canadienne de Génie Civil (Can. J. Civ. Eng.)* Vol. 36, p. 1144–1154.
- [3] Kirkpatrick S., Gelatt C. D. and Vecchi M. P.(1983), Optimization by simulated annealing, *Science* 13 May 1983, Volume 220, Number 4598, p. 671-680.
- [4] Faure J-B. (2006), Use of simulated annealing optimization for sensitivity analysis in fluvial hydraulics, HydroInformatics 2006, Nice, France.
- [5] Faure J-B., Dhervillez L. and Vidal J-P, (2002) Automatic evaluation of the uncertainty of the surface of a flooded area, HydroInformatics 2002, Cardiff, UK.
- [6] Giraud F.M., Faure J-B., Zimmer D., Lefevre J.C. and Skaggs R.W., (1997) Hydrologic modeling of a complex wetland, *Journal of Irrigation and Drainage Engineering*, Vol. 123, No. 5, September/October, p. 344-353.