



Assimilation de données pour les modèles d'hydraulique fluviale. Estimation de paramètres, analyse de sensibilité et décomposition

Cyril Mazauric

► To cite this version:

Cyril Mazauric. Assimilation de données pour les modèles d'hydraulique fluviale. Estimation de paramètres, analyse de sensibilité et décomposition. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 2003. Français. NNT : . tel-00004632

HAL Id: tel-00004632

<https://theses.hal.science/tel-00004632>

Submitted on 11 Feb 2004

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.

Thèse

présentée par

Cyril MAZAURIC

Pour obtenir le titre de
Docteur de l'Université Joseph Fourier - Grenoble I
(arrêtés ministériels du 5 juillet 1984 et du 30 mars 1992)

Spécialité : **Mathématiques Appliquées**

**Assimilation de données
pour les modèles d'hydraulique fluviale.
Estimation de paramètres, analyse de sensibilité
et décomposition de domaine.**

Date de soutenance : 12 Décembre 2003

Composition du jury :

M. Philippe Bois	Président
M. Denis Dartus	Rapporteur
M. Dan NGuyen	Rapporteur
M. François-Xavier Le Dimet	Directeur
M. Philippe Belleudy	Co-directeur
M. Jean-Baptiste Faure	Examineur
M. NGuyen Van Diep	Invité

Thèse préparée au sein du Laboratoire de Modélisation et Calcul (IMAG
projet IDOPT) et de l'INRIA rhônes-alpes.

Remerciements

Cette thèse est le résultat de trois années de travail, je remercie ici toutes les personnes qui m'ont épaulées.

Tout d'abord, je tiens à remercier les membres du jury et en premier lieu Philippe Bois qui a accepté de le présider.

Merci à Denis Dartus et Dan NGuyen qui ont accepté d'être les rapporteurs de ce travail. Je remercie aussi Jean-Baptiste Faure d'avoir accepté la charge d'être examinateur de ce travail.

Enfin et surtout je tiens à remercier sincèrement mes directeurs de thèse qui ont su m'aider et me soutenir dans les bons moments comme dans les pires. François-Xavier Le Dimet qui grâce à ses nombreux conseils m'a fait découvrir les finesses et les possibilités de l'assimilation de données variationnelle et Philippe Belleudy qui a réussi à m'ouvrir au monde particulier de l'hydraulique fluviale.

Je remercie aussi tous ceux qui au long de ces trois ans m'ont accordés leurs amitiés mais aussi qui ont eu la patience de lire et relire ce mémoire de thèse. Tout d'abord et par ordre d'apparence dans ma vie, mon collègue de bureau William qui m'a soutenu lors de mes moments de doutes et qui durant nos longues conversations m'a apporté ses compétences et son amitié fidèle. Je remercie aussi Claire et Céline qui par leurs bonnes humeurs et leurs conseils éclairés ont rendus la vie au LMC plus agréable.

Je remercie enfin et surtout celle qui partage ma vie tout simplement pour être elle-même ...

Table des matières

1	Introduction	11
2	Assimilation de données et hydraulique	17
2.1	Assimilation de données variationnelle et estimation de paramètres	18
2.1.1	Définition de l'assimilation de données variationnelle	18
2.1.2	Algorithme de résolution	21
2.1.3	Contrôle des conditions aux limites	21
2.1.4	Contrôle d'un paramètre	24
2.2	Assimilation de données et Analyse de sensibilité	26
2.2.1	Illustration de l'importance du second ordre	26
2.2.2	Analyse de sensibilité et second ordre	28
2.2.3	Algorithme de calcul de la sensibilité	29
2.3	Conclusion	33
3	Les modèles d'hydraulique	35
3.1	Un modèle d'hydraulique fluviale monodimensionnel : Carima	35
3.1.1	Présentation du logiciel Carima	35
3.1.2	Les équations de Saint Venant	38
3.1.3	Discrétisation des équations de Saint Venant	39
3.1.4	Casiers	43
3.1.5	Conditions aux limites	43
3.1.6	Résolution du système des équations discrétisées	44
3.1.7	Étude de la stabilité	44
3.1.8	Carima en quelques chiffres	45
3.1.9	Construction du modèle Loire	45
3.1.10	Conclusion	50
3.2	Une maquette de calcul des écoulements bidimensionnel : LMCFLD	51
3.2.1	Les équations de Saint Venant	51
3.2.2	Les conditions aux limites	52
3.2.3	Changement des variables pour les équations de Saint-Venant	52
3.2.4	Discrétisation des équations de Saint Venant	53
3.2.5	Résolution du système des équations discrétisées	57
3.2.6	Mise en place du modèle	59
3.2.7	LMCFLD en quelques chiffres	61
3.2.8	Conclusion	62

4	Mise en œuvre de l'assimilation de données : techniques et difficultés	63
4.1	La différentiation automatique	63
4.1.1	La théorie	64
4.1.2	La pratique	66
4.1.3	Les difficultés à contourner	70
4.1.4	Modification manuelle du code d'Odyssée	72
4.1.5	Validation du code Adjoint d'Odyssée	74
4.1.6	Le second ordre	76
4.1.7	Conclusion	80
4.2	Expériences jumelles	80
4.3	Le code de minimisation M1QN3	81
4.3.1	Description de la méthode	81
4.3.2	Utilisation du code	82
4.4	Algorithme pour l'estimation du paramètre K	83
4.4.1	Algorithme de la subroutine principale	83
4.4.2	Algorithme de la subroutine SIMUL	84
4.4.3	Algorithme de la subroutine Flux	84
4.5	Conclusion	85
5	Assimilation de données et prévision	87
5.1	Estimation de la condition initiale	88
5.1.1	Carima	88
5.1.2	LMCFLD	94
5.2	Estimation des conditions aux limites	100
5.2.1	Trois exemples de condition aux limites	100
5.2.2	Influence du relief dans l'estimation de la condition aux limites . . .	100
5.3	Couplage CI et CL	103
5.4	Conclusion	109
6	Assimilation de données et identification de paramètres	111
6.1	Estimation de la rugosité	112
6.1.1	Un simple bassin	112
6.1.2	Un jeu de Manning binaire	115
6.1.3	Influence du relief	115
6.1.4	Schématisation du jeu de Manning	118
6.1.5	Loire	128
6.2	Estimation du relief équivalent	133
6.2.1	La colline	133
6.2.2	La route	139
6.2.3	Loire	149
6.3	Conclusion	152
7	Analyse de sensibilité	153
7.1	Sensibilité par rapport à la perturbation des rugosités	153
7.1.1	Calcul de la sensibilité à la perturbation des rugosités sur la Loire .	153
7.1.2	Utilisation de la sensibilité pour la schématisation de la rugosité . .	155
7.2	Sensibilité par rapport aux observations	160
7.2.1	Un exemple trivial	160

7.2.2	Une illustration de la nécessité de bien placer les capteurs	160
7.3	Conclusion	161
8	Assimilation de données et parallélisation	163
8.1	Décomposition de domaine	164
8.2	Mise en œuvre de la méthode	166
8.2.1	Modèle général	166
8.2.2	Hardware	167
8.2.3	Parallélisation	167
8.3	Une décomposition régulière	168
8.3.1	Un exemple avec un jeu d'observation complet	168
8.3.2	Un exemple avec un jeu d'observation dégradé	170
8.4	Une décomposition irrégulière	171
8.5	Un exemple de rivière dont le relief et la rugosité sont plus "chaotique" . .	173
8.6	Vers un critère d'arrêt de la méthode itérative	174
8.7	Conclusion	176
9	Conclusion	177
A	Quelques notions mathématiques	181

Contexte de l'étude

Ce travail de thèse s'inscrit dans le cadre du projet Sino-Européen ANFAS (data fusion for flood analysis and decision support). Le but de ce projet est de créer un système d'aide à la planification préventive des inondations par la modélisation hydraulique et la cartographie assistées via Internet.

ANFAS est un projet de 3 ans (janvier 2000- décembre 2002) cofinancé par l'union Européenne (programme IST) et par le gouvernement Chinois.

ANFAS est un système d'aide à la décision qui s'adresse principalement aux services techniques en appui aux décideurs pour la planification préventive des inondations. Il a pour objectif de mettre à leur disposition, via une interface Internet, un outil de construction, de simulation, et de comparaison, de différents scénarii d'inondation de plaine.

C'est un système distribué basé sur la technologie du Web, qui intègre et relie (de façon transparente pour l'utilisateur) les différents modules : base de données gérée sous SIG, modèles hydrauliques, méthodologie d'estimation de dommages.

Des données d'origines et de natures très diverses (imagerie satellite, photo interprétation, modèles numériques de terrain, données hydrologiques) sont intégrées dans les bases de données pour être utilisées dans les différents modules : construction et calage des modèles hydrauliques, évaluation des éléments à risque. Le système inclut des fonctionnalités qui permettent à l'utilisateur de visualiser les résultats de ces scénarii sur le Web, et/ou sur son propre micro. ANFAS est testé sur trois sites pilotes : la rivière Vah en Slovaquie, la Loire moyenne en France, et une portion du Yangtzé en Chine.

Chapitre 1

Introduction

Les inondations de plaine qu’a connues l’Europe ces dernières années nous rappellent que leurs conséquences peuvent être catastrophiques : les biens exposés dans les plaines alluviales villes, entreprises, réseaux majeurs, patrimoine historique sont nombreux et sont autant d’éléments à forte valeur. De plus, la surface sinistrée est généralement importante car les fleuves peuvent avoir un long cours associé à une vallée alluviale large avec un faible gradient topographique. Pour les aménageurs ou les gestionnaires du risque, la prévision fiable des crues de plaine présente un intérêt évident.

Après avoir poussé l’urbanisation des cités en construisant un peu partout, sans respecter ni le lit naturel des rivières ni les zones d’expansions, la nécessité d’étudier et de préciser les aléas du risque inondation a conduit à développer les méthodes de simulation numérique de la propagation des crues. Ce travail concerne l’amélioration des modèles numériques utilisés de manière opérationnelle par les ingénieurs dans l’étude des inondations.

La calcul de prévisions fiables des inondations ne peut se concevoir sans l’utilisation de toutes les informations disponibles. Les équations mathématiques qui permettent la construction d’un modèle, les images satellites par exemple qui décrivent le domaine, mais aussi les mesures faites *in situ*. Ces dernières données, appelées observations, sont nécessaires à la bonne qualité des prévisions puisqu’elles sont le seul témoin de la réalité du terrain.

Généralement, ces observations ne sont utilisées que manuellement afin de caler partiellement le modèle. L’assimilation de données variationnelle que nous introduirons dans le deuxième chapitre est une méthode mathématique qui permet d’intégrer automatiquement les observations dans le modèle. Basée sur la théorie du contrôle optimal, cette méthode permet d’améliorer la prévision (voir figure 1.1) grâce à la minimisation d’une fonction coût mesurant la différence entre les observations et les prévisions.

Les applications de la méthode de l’adjoint (c’est-à-dire l’assimilation de données variationnelle) sont nombreuses. Dans ce deuxième chapitre nous développerons la théorie de l’assimilation de données appliquées à l’estimation de certains paramètres afin de minimiser l’écart entre le calcul du modèle et les observations. Ainsi nous développerons la théorie de l’assimilation de données en vue d’estimer la condition initiale, la condition aux limites et divers paramètres comme la rugosité ou le relief.

Ensuite nous aborderons la question de l’analyse de sensibilité grâce à la résolution du système adjoint au second ordre.

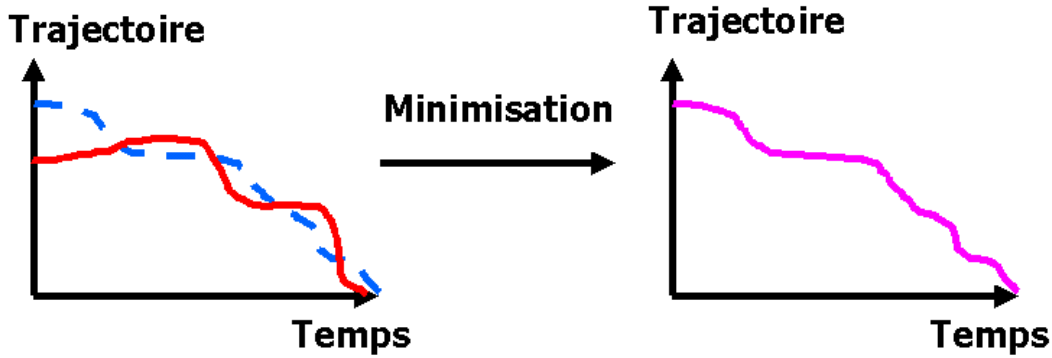


FIG. 1.1 – *Illustration de l'effet de l'assimilation de données variationnelle sur le calcul de la trajectoire du modèle. Dans la figure de gauche la courbe discontinue (bleue) décrit la trajectoire idéale du modèle et la courbe continue (rouge) décrit la trajectoire calculée par le modèle, ces deux dernières sont assez éloignées. La figure de droite montre qu'après minimisation, i.e. mise en place de la méthode de l'assimilation de données, la trajectoire calculée rejoint la trajectoire idéale. Les cercles verts désignent la position des observations disponible. La "trajectoire" est l'évolution de l'état du système (les valeurs des variables d'état en tout point du domaine) en fonction du temps.*

Le troisième chapitre sera consacré à l'étude des modèles d'hydraulique fluviale qui seront utilisés pour tester l'assimilation de données. Nous verrons d'abord un code de calcul professionnel et opérationnel, Carima. Ce modèle, basé sur la résolution des équations de Saint Venant monodimensionnelles est distribué par la SOGREAH. Dans les limites des informations publiées et des besoins de notre démonstration, nous présenterons les équations et la discrétisation utilisée pour calculer l'écoulement. Ensuite nous nous pencherons sur la présentation d'un logiciel d'hydraulique fluviale bidimensionnel LMCFLD.

Mais la mise en œuvre de l'assimilation de données variationnelle n'est pas une chose aisée. Le quatrième chapitre est consacré à la présentation des différentes étapes de la mise en place de ces méthodes : la création du code adjoint grâce à un logiciel de différentiation automatique (les règles d'écriture du code et les pièges à éviter seront exposés), l'utilisation de la méthode des expériences jumelles afin de pallier au problème du manque d'observations, l'utilisation d'un code de minimisation opérationnel, M1QN3, afin de minimiser la fonction coût puis l'écriture d'un algorithme chargé d'estimer un paramètre.

Afin de calculer une prévision fiable, il faut partir sur des bases saines, c'est-à-dire qu'il faut donner une condition initiale optimale (voir figure 1.2). Dans la première partie du chapitre cinq, nous verrons des exemples de l'estimation de la condition initiale avec les codes de calcul Carima et LMCFLD. La deuxième partie de ce chapitre sera consacrée à la présentation de quelques exemples sur l'estimation de la condition aux limites avec le logiciel LMCFLD. En effet, l'estimation de cette variable est non seulement primordiale pour faire des prévisions fiables, mais il est aussi très important de pouvoir l'estimer après un événement. Ainsi après une inondation, une multitude de traces peuvent se mesurer sur les murs des maisons ou sur le tronc des arbres, ces observations permettent de préciser les caractéristiques de la crue responsable de l'inondation.

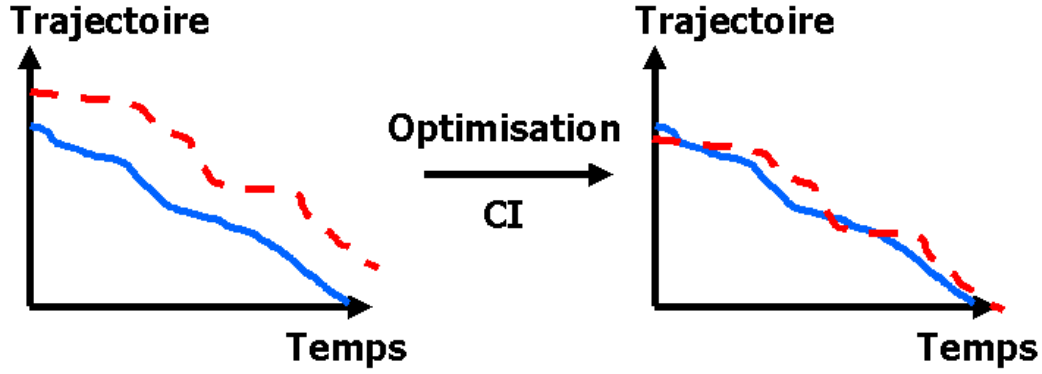


FIG. 1.2 – Illustration de l'importance de l'estimation de la condition initiale. Dans la figure de gauche, la courbe continue (bleue) est la trajectoire idéale et la courbe discontinue (rouge) est la trajectoire calculée. Après estimation d'une condition initiale optimale, la figure de droite illustre le rapprochement entre la trajectoire calculée et idéale.

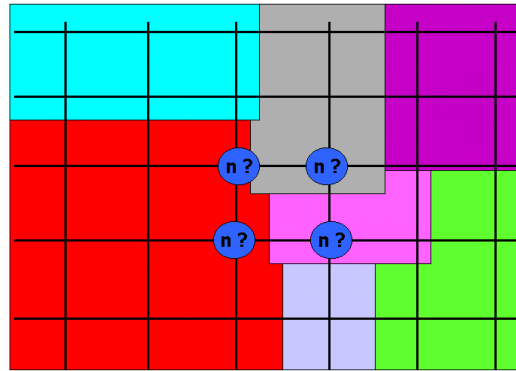


FIG. 1.3 – Illustration du problème posé par la détermination d'un jeu de Manning optimal à partir d'une distribution des rugosités complexe. Comment ne pas perdre l'information contenue dans les mailles ?

Il faut maintenant s'intéresser au problème du calage du modèle en définissant les paramètres de rugosité et de relief.

La rugosité représente la nature de la surface du sol et est généralement estimée par l'expertise à partir d'observations du terrain ou, sur de grands domaines complexes à partir d'une imagerie aérienne ou satellitale. Mais cette donnée très précise ne peut être utilisée sans précautions. En effet comme l'illustre la figure 1.3, si on donne à chaque nœud du maillage la valeur réelle de la rugosité, on ne tient pas compte de la rugosité à l'intérieur de la maille et par conséquent on apporte une erreur supplémentaire dans le calcul de l'écoulement. C'est pourquoi on doit estimer un jeu de rugosité équivalent qui va permettre de calculer un écoulement fiable.

L'estimation de la topographie pose le même type de problème. Si l'on utilise la topographie exacte à chaque nœud du maillage, on prend le risque de passer à côté d'un pic du relief très influent sur le comportement de l'écoulement (voir la figure 1.4).

Mais il n'y a pas que la topographie de la plaine inondable qui doit être étudiée avec

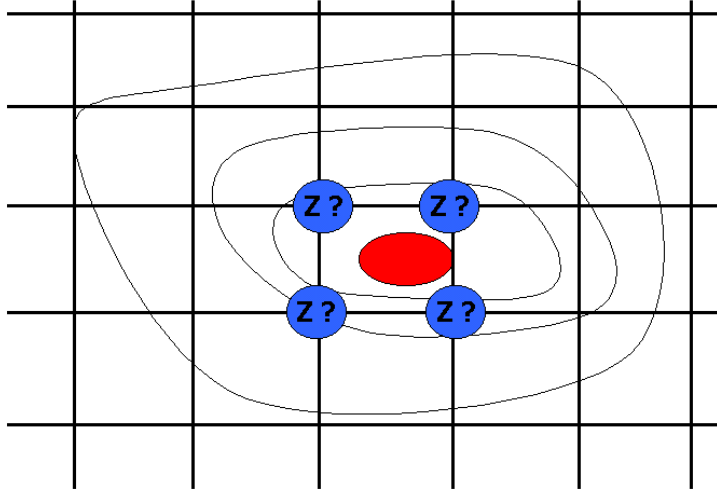


FIG. 1.4 – *Illustration du problème posé par la détermination d'une topographie optimale à partir d'une topographie du domaine plus précise que celle nécessaire. Comment ne pas perdre l'information contenue dans les mailles ?*

précautions. Ainsi, comment définir la topographie de la rivière alors que celle-ci est constamment noyée. La seule solution est d'utiliser les sections mesurées par un géomètre qui permettent de mettre en place un modèle monodimensionnel. Comme l'illustre la figure 1.5, elles ne sont pas suffisantes pour connaître la topographie en chaque point du maillage bidimensionnel. Généralement, une interpolation linéaire est appliquée afin de calculer ces sections manquantes mais ces interpolations ne tiennent pas forcément compte de tous les paramètres. Par conséquent, comment définir une topographie équivalente dans le lit de la rivière ?

Enfin, certaines rivières à fort potentiel sédimentaire changent constamment de géométrie. Les mesures effectuées par un géomètre étant très onéreuses, il est financièrement impossible de mettre à jour cette géométrie.

Pour toutes ces raisons, l'estimation de la topographie doit être menée de manière précise et judicieuse. Ainsi, comme nous le verrons dans la deuxième partie du sixième chapitre à travers quelques exemples, l'assimilation de données variationnelle apporte une solution à l'estimation d'une topographie équivalente.

Cependant, quelle que soit la variable que l'on cherche à estimer, il est nécessaire d'en connaître l'influence sur le comportement de l'écoulement. Déterminer les zones de plus fortes influences, c'est connaître les zones où l'estimation doit être la plus précise et donc la plus minutieusement menée. Dans le septième chapitre nous allons mener une analyse de sensibilité afin de déterminer l'influence du jeu de rugosité sur l'écoulement ce qui permettra de schématiser ce jeu de données.

Nous verrons ensuite que l'analyse de sensibilité va nous permettre de déterminer les zones où les observations sont le plus nécessaires à une estimation de variable optimale. Ce qui permettra de déterminer la position des capteurs nécessaires à une bonne estimation de paramètres.

L'étude hydraulique d'un domaine peut nécessiter plusieurs niveaux d'expertises. Une

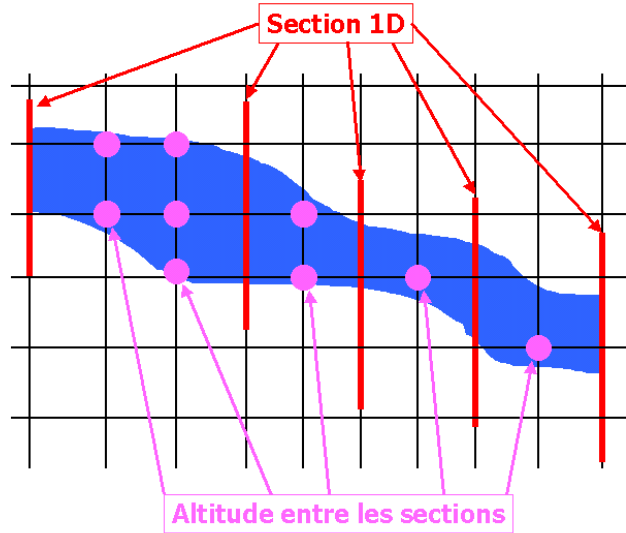


FIG. 1.5 – Illustration du problème posé pour obtenir une topographie bidimensionnelle de la rivière à partir de la connaissance des quelques sections.

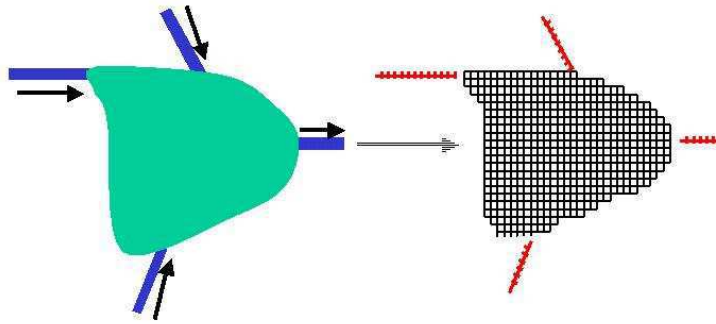


FIG. 1.6 – Illustration de l'utilité de la méthode de l'assimilation de données variationnelle pour le couplage de modèles d'hydraulique fluvial 1d et 2D.

connaissance globale de l'écoulement sur le domaine entier et une connaissance plus précise de l'écoulement sur une petite partie de celui-ci (voir figure 1.6). Ce "zoom" imposé par le type de l'étude ne peut se concevoir sans un couplage de deux modèles. Un modèle monodimensionnel pour calculer l'écoulement global dans le domaine et un modèle bidimensionnel pour calculer l'écoulement dans la partie où la précision est requise. Le problème est alors de savoir quelles valeurs doit-on communiquer d'un modèle à l'autre.

Pour les grands domaines et pour des raisons de rapidité des résultats requis dans le cas d'une étude en temps réel, la parallélisation des codes va devoir être envisagé. Dans le chapitre huit, nous allons étudier l'utilisation des méthodes d'assimilation de données dans le cadre d'une décomposition de domaine afin de contrôler les interfaces entre les sous domaines.

Chapitre 2

Assimilation de données et hydraulique

La modélisation des phénomènes physiques confronte l'utilisateur à de nombreux problèmes liés à la précision des résultats. La construction d'un modèle est basée sur la connaissance des deux sources d'informations que sont les modèles (physiques ou numériques) et les observations. Ces deux types de données apportent leurs lots d'approximations. En effet, les modèles ne sont qu'une représentation simplifiée d'une réalité physique complexe et les observations, trop localisées, ne permettent pas une connaissance satisfaisante du domaine considéré.

Les observations effectuées sur une rivière le sont bien souvent uniquement sur les hauteurs d'eau grâce à des règles positionnées dans le lit.

Le problème se pose donc en ces termes : *Comment intégrer les observations dans les modèles tout en sachant que ceux-ci sont entachés d'erreurs dues à la discrétisation et que les observations sont imprécises à cause des erreurs de mesures ?*

Les techniques d'assimilation de données permettent de combiner de manière optimale les informations fournies par le modèle et celles provenant des observations, quelles soient d'origine statistique ou directe, afin de reconstruire le plus fidèlement possible l'état du système. La fidélité est une notion qui, ici, est à définir en fonction des techniques employées ou du but de l'étude considérée.

Ces méthodes se décomposent en trois grandes familles :

- *L'interpolation* : sur le domaine considéré, la grille de discrétisation marque les points où l'évolution des variables physiques va être calculée. Les observations faites sur la zone ne correspondent pas forcément avec les nœuds de ce maillage (en cas d'utilisation d'un schéma aux différences finies par exemple). Afin d'estimer l'état du système une simple interpolation linéaire est en général utilisée, mais celle-ci n'étant fondée sur aucune justification physique, la précision des résultats est plus que discutable.
- *Le filtrage* : cette approche permet d'obtenir une estimation du meilleur état présent du système à partir de toutes les informations passées et présentes. Elle consiste à corriger, à partir d'un champ d'observations sur une période donnée, la trajectoire du modèle durant son intégration afin d'obtenir une condition initiale (ou aux limites) optimale. Cette méthode (basée sur le filtre de Kalman) nécessite l'utilisation des matrices de covariance du système. La manipulation de ces matrices, généralement

de très grandes dimensions, rend la méthode très dépendante de la puissance des machines. Cette technique est aussi appelée : *assimilation de données a.*

- *Le lissage* : inspirée de la théorie du contrôle optimal, cette démarche permet de fournir, pour une période de temps donnée, une estimation de l'évolution du système qui s'ajuste optimalement aux observations et au modèle utilisé. C'est un outil pertinent pour le calage des modèles, l'analyse de sensibilité et l'estimation de l'erreur du modèle.

L'optimisation est réalisée par une méthode de gradient dont la valeur est estimée grâce aux méthodes adjointes introduites par Le Dimet et Talagrand [11]. Plusieurs itérations du code d'optimisation sont généralement nécessaires afin de calculer le gradient. Le coût de calcul d'une telle méthode peut alors vite devenir très important. De plus cette approche nécessite l'écriture de l'adjoint du modèle direct qui est relativement difficile pour les codes de calcul complexes.

Cette technique est aussi appelée : *assimilation de données variationnelle*.

Le travail décrit dans ce mémoire de thèse est basé sur l'étude des méthodes d'assimilation de données variationnelle en hydraulique fluviale. Ce chapitre sera donc focalisé sur la compréhension de ces méthodes. Dans un premier temps, nous développerons la théorie de l'assimilation de données dédiée au contrôle de différentes variables puis nous verrons l'utilisation de cette méthode dans le but de faire une analyse de sensibilité de diverses variables de contrôle par rapport à la perturbation d'un paramètre.

2.1 Assimilation de données variationnelle et estimation de paramètres

Depuis quelques années les techniques d'assimilation de données variationnelle sont opérationnelles en météorologie et en océanographie. Par exemple Météo-France utilise ces méthodes depuis l'an 2000.

Le principe de ces méthodes est de minimiser une fonctionnelle mesurant l'écart entre la prévision et les observations. Cette minimisation est basée sur le contrôle d'une variable (condition aux limites, initiale ou variables empiriques tels que le relief ou la rugosité dans le cas de l'hydraulique).

En hydraulique fluviale, l'assimilation de données est pour l'instant très peu utilisée de manière opérationnelle. FLDWAV [6] et MIKE21 ont mis en place l'assimilation de données stochastique afin de caler leurs modèles, mais pour l'instant aucune approche variationnelle n'est disponible dans un logiciel de modélisation hydraulique professionnel. Dans le domaine de la recherche, on peut signaler les travaux de Yang [24] sur l'assimilation de données variationnelle et la sédimentation des rivières, ceux de Ngnepieba [17] sur l'hydrologie et les travaux de Losch [14] sur l'estimation du relief équivalent.

2.1.1 Définition de l'assimilation de données variationnelle

Afin d'étudier les méthodes d'assimilation de données, nous allons considérer dans un premier temps que la variable de contrôle de notre étude est la condition initiale, U . Nous verrons ultérieurement le développement de ces techniques par rapport à d'autres variables

de contrôle.

Soit le modèle général suivant :

$$\begin{cases} \frac{dX}{dt} + F(X) = 0 \\ X(0) = U \end{cases} \quad (2.1)$$

où F est un opérateur d'équations aux dérivées partielles (éventuellement non linéaire), X est la variable d'état du système et U la condition initiale, variable de contrôle de notre étude. Ce système sera résolu sur l'intervalle de temps $[0, T]$.

Soit X_{obs} , un jeu de données que nous considérerons continu en temps, χ l'espace des prévisions tel que $X \in \chi$ et χ_{obs} celui des observations tel que $X_{obs} \in \chi_{obs}$.

Soit l'opérateur C projetant une variable de l'espace des prévisions sur celui des observations :

$$C : \chi \rightarrow \chi_{obs}$$

Comparons maintenant mesures et observations grâce à la fonctionnelle J que nous cherchons à minimiser :

$$J(U) = \frac{1}{2} \|CX(U) - X_{obs}\|_{\chi_{obs}}^2 + \frac{1}{2} \|U - U_0\|_{\mathcal{U}}^2 \quad (2.2)$$

avec $\|\cdot\|_{\chi_{obs}}^2$ la norme définie dans l'espace des observations et $\|\cdot\|_{\mathcal{U}}^2$ celle de l'espace dans lequel se trouve les conditions initiales possibles. U_0 est la valeur *a priori* de la condition initiale.

Remarque : La fonction coût n'est pas obligatoirement convexe ; par conséquent elle peut avoir plusieurs minima locaux qui sont autant de résultats possibles pour la minimisation de J . La connaissance *a priori* d'une condition initiale proche de la condition optimale nous permet d'éviter ce problème de minima locaux. En effet, le deuxième terme de la fonction coût $\|U - U_0\|_{\mathcal{U}}^2$ va diriger la minimisation vers la valeur optimale où se trouve le minimum global. Toute la difficulté de ce terme de forçage est d'estimer au mieux une valeur *a priori* de la condition initiale.

Remarque : A travers la définition de la norme utilisée dans la fonction coût, on peut prendre en compte toutes les informations disponibles. Par exemple, la matrice de covariance de l'erreur peut être utilisée dans la définition de cette norme :

$$\|X\|_{\chi_{obs}}^2 = \langle AX, X \rangle$$

Condition d'optimalité et modèle adjoint

Le problème est donc de trouver U^* minimisant J et caractérisé par la condition d'Euler :

$$\nabla_U J(U^*) = 0 \quad (2.3)$$

En pratique U^* est calculé grâce à un algorithme d'optimisation du type gradient conjugué.

Le but de la méthode est donc de calculer le **gradient de J**.

2.1. ASSIMILATION DE DONNÉES VARIATIONNELLE ET ESTIMATION DE PARAMÈTRES

Soit la Gâteaux dérivée¹ du modèle (2.1) suivant la direction δU de U :

$$\begin{cases} \frac{d\widehat{X}}{dt} + \left[\frac{\partial F(X)}{\partial X} \right] \cdot \widehat{X} &= 0 \\ \widehat{X}(0) &= \delta U \end{cases} \quad (2.4)$$

avec \widehat{X} la dérivée directionnelle de X suivant δU .

Soit la Gâteaux dérivée de la fonction coût (2.2)

$$(\nabla_U J, \delta U) = (C.X - X_{obs}, C\widehat{X}) \quad (2.5)$$

Soit p , la variable adjointe (que nous définirons ultérieurement), une variable de même dimension que \widehat{X} . Cette variable est multipliée au sens du produit scalaire avec le modèle (2.4).

$$\left(\frac{d\widehat{X}}{dt}, p \right) + \left(\left[\frac{\partial F(X)}{\partial X} \right] \cdot \widehat{X}, p \right) = 0 \quad (2.6)$$

Après une intégration par parties (IPP)² sur le premier terme de l'équation précédente, nous obtenons :

$$(\widehat{X}(T), p(T)) - (\widehat{X}(0), p(0)) - \left(\frac{dp}{dt}, \widehat{X} \right) + \left(\left[\frac{\partial F(X)}{\partial X} \right] \cdot \widehat{X}, p \right) = 0 \quad (2.7)$$

Grâce à la définition suivante,

$$(AX, Y) = (X, A^t Y) \quad (2.8)$$

l'équation (2.7) devient :

$$(\widehat{X}(T), p(T)) - (\delta U, p(0)) + \left(-\frac{dp}{dt} + \left[\frac{\partial F(X)}{\partial X} \right]^t p, \widehat{X} \right) = 0 \quad (2.9)$$

En réécrivant (2.5), nous pouvons dire que la Gâteaux dérivée³ de J suivant δU est de la forme :

$$(\delta U, \nabla_U J) - (C^t (C.X - X_{obs}), \widehat{X}) = 0 \quad (2.10)$$

Afin de calculer le gradient de J , nous devons exhiber la linéarité de ce gradient par rapport à δU . Pour ce faire posons p solution de

$$\begin{cases} \frac{dp}{dt} - \left[\frac{\partial F}{\partial X} \right]^t p &= C^t (C.X - X_{obs}) \\ p(T) &= 0 \end{cases} \quad (2.11)$$

En comparant (2.9) et (2.10), nous obtenons :

$$\nabla_U J = -p(0) \quad (2.12)$$

¹voir Annexe A

²voir annexe A

³voir annexe A

Cette équation nous donne une caractérisation du minimum de J en fonction de la variable adjointe p .

Par conséquent, la minimisation de J passe par la résolution du système d'optimalité suivant :

$$\left\{ \begin{array}{lcl} \frac{dX}{dt} + F(X) & = & 0 \\ X(0) & = & U \\ \frac{dp}{dt} - \left[\frac{\partial F}{\partial X} \right]^t p & = & C^t (C.X - X_{obs}) \\ p(T) & = & 0 \\ \nabla_U J & = & -p(0) \end{array} \right. \quad (2.13)$$

2.1.2 Algorithme de résolution

Afin de calculer le minimum de J , nous allons proposer une approche basée sur un algorithme de descente, valable quelle que soit la variable de contrôle V considérée :

1. on se donne une valeur initiale de la variable à estimer : V_0
2. pour chaque itération ($n > 0$)
 - on résout le système direct avec V_n ,
 - on résout le modèle adjoint afin de calculer $\left[\frac{\partial F}{\partial X} \right]^t$, puis on résout le système adjoint. Cette résolution nous permet d'obtenir une valeur du gradient de J ,
 - on calcule le pas de descente ρ_n et la direction de descente, D_n .
 - il vient

$$V_{n+1} = V_n + \rho_n D_n$$

Remarque : Le calcul de la direction de descente dans l'algorithme précédent peut être envisagé de manière différente. Dans ce travail, nous avons fait le choix d'utiliser le code d'optimisation créé par LeMarechal et al [12], inspiré de la méthode quasi-Newton dite de BFGS.

Nous avons vu précédemment le calcul du gradient de la fonction coût par rapport à la condition initiale. Mais d'autres variables de contrôle peuvent être prises en compte. Nous allons voir, dans la suite, le détail du calcul du gradient de la fonction coût, successivement par rapport aux conditions aux limites puis par rapport au modèle numérique (par exemple la rugosité ou le relief).

Concrètement, la mise en oeuvre de l'assimilation de données nécessite un (gros) travail d'écriture du modèle adjoint que nous étudierons plus tard. Cependant, quel que soit la variable de contrôle, le système adjoint est le même. Ainsi, une fois passé le stade difficile de l'écriture du code adjoint, ce même code servira quel que soit le but de l'étude.

2.1.3 Contrôle des conditions aux limites

Après les crues du Gard en 2002, seules les "laisses de crues", c'est-à-dire le relevé des niveaux maximum atteints par l'inondation sur les murs des maisons par exemple, témoigne du passage de l'eau. A partir de ces informations on cherche à déterminer l'hydrogramme entrant dans le domaine (i.e. le débit amont qui a provoqué cette inondation).

2.1. ASSIMILATION DE DONNÉES VARIATIONNELLE ET ESTIMATION DE PARAMÈTRES

L'estimation des conditions aux limites amont et aval est donc très important. Dans cette section nous allons donc développer la théorie de l'assimilation de données dédiée à l'estimation de cette variable.

Soit le système général discrétisé suivant :

$$\begin{cases} \frac{dX}{dt} + F(X) + D_1.CL_1 + D_2.CL_2 = 0 \\ X(0) = 0 \end{cases} \quad (2.14)$$

où F est la discrétisation d'un opérateur d'équations aux dérivées partielles, X est la variable d'état du système, CL_1 la condition aux limites amont et CL_2 celle définie en aval, ce seront les variables de contrôle de notre étude. Ce système sera résolu sur l'intervalle de temps $[0, T]$

Soit X_{obs} , un jeu de données que nous considérerons continu en temps, χ l'espace des prévisions tel que $X \in \chi$ et χ_{obs} celui des observations tel que $X_{obs} \in \chi_{obs}$.

Soit l'opérateur C projetant une variable de l'espace des prévisions sur celui des observations :

$$C : \chi \rightarrow \chi_{obs}$$

Comparons maintenant mesures et observations grâce à la fonctionnelle :

$$J(CL_1, CL_2) = \frac{1}{2} \|CX(CL_1, CL_2) - X_{obs}\|_{\chi_{obs}}^2 + \frac{1}{2} \|CL_1 - CL_1^0\|_{\mathcal{CL}}^2 + \frac{1}{2} \|CL_2 - CL_2^0\|_{\mathcal{CL}}^2 \quad (2.15)$$

avec $\|\cdot\|_{\chi_{obs}}^2$ la norme définie dans l'espace des observations et $\|\cdot\|_{\mathcal{CL}}^2$ celle de l'espace dans lequel se trouvent les conditions aux limites possibles.

CL_1^0 (resp. CL_2^0) est la valeur *a priori* de la condition aux limites amont (resp. aval).

Le problème est de trouver CL_1^* et CL_2^* minimisant J , cette minimisation est caractérisée par la condition d'Euler :

$$\begin{cases} \nabla_{CL_1} J(CL_1^*) = 0 \\ \nabla_{CL_2} J(CL_2^*) = 0 \end{cases} \quad (2.16)$$

En pratique CL_1^* et CL_2^* sont calculés par un algorithme d'optimisation du type gradient conjugué.

Le but de la méthode est de calculer le **gradient de J**

Soit la Gâteaux dérivée⁴ du modèle (2.14) suivant les directions δCL_1 de CL_1 et δCL_2 de CL_2 :

$$\begin{cases} \frac{d\widehat{X}}{dt} + \left[\frac{\partial F(X)}{\partial X} \right] \cdot \widehat{X} + D_1.\delta CL_1 + D_2.\delta CL_2 = 0 \\ \widehat{X}(0) = 0 \end{cases} \quad (2.17)$$

avec \widehat{X} la dérivée directionnelle de X suivant δCL_1 et δCL_2 .

Soit la Gâteaux dérivée de la fonction coût (2.15)

$$(\nabla_{CL_1} J, \delta CL_1) + (\nabla_{CL_2} J, \delta CL_2) = (C.X - X_{obs}, C\widehat{X}) \quad (2.18)$$

⁴voir annexe A

Soit p , la variable adjointe, une variable de même dimension que \widehat{X} . Cette variable est multipliée au sens du produit scalaire avec le modèle (2.17).

$$\left(\frac{d\widehat{X}}{dt}, p\right) + \left(\left[\frac{\partial F(X)}{\partial X}\right] \cdot \widehat{X}, p\right) + (D_1 \cdot \delta CL_1, p) + (D_2 \cdot \delta CL_2, p) = 0 \quad (2.19)$$

Après une intégration par parties⁵ sur le premier terme de l'équation précédente, nous obtenons :

$$\begin{aligned} (\widehat{X}(T), p(T)) - (\widehat{X}(0), p(0)) - \left(\frac{dp}{dt}, \widehat{X}\right) + \left(\left[\frac{\partial F(X)}{\partial X}\right] \cdot \widehat{X}, p\right) \\ + (D_1 \cdot \delta CL_1, p) + (D_2 \cdot \delta CL_2, p) = 0 \end{aligned} \quad (2.20)$$

avec la définition suivante,

$$(AX, Y) = (X, A^t Y) \quad (2.21)$$

l'équation (2.20) devient :

$$(\widehat{X}(T), p(T)) + \left(-\frac{dp}{dt} + \left[\frac{\partial F(X)}{\partial X}\right]^t p, \widehat{X}\right) + (\delta CL_1, D_1^t \cdot p) + (\delta CL_2, D_2^t \cdot p) = 0 \quad (2.22)$$

Soit la Gâteaux dérivée⁶ de J suivant δCL_1 et δCL_2 , une variante de (2.18)

$$(\delta CL_1, \nabla_{CL_1} J) + (\delta CL_2, \nabla_{CL_2} J) - (C^t (C \cdot X - X_{obs}), \widehat{X}) = 0 \quad (2.23)$$

Posons p solution de

$$\begin{cases} \frac{dp}{dt} - \left[\frac{\partial F}{\partial X}\right]^t p = C^t (C \cdot X - X_{obs}) \\ p(T) = 0 \end{cases} \quad (2.24)$$

En comparant (2.22) et (2.23), nous obtenons :

$$\begin{cases} \nabla_{CL_1} J = D_1^t \cdot p \\ \nabla_{CL_2} J = D_2^t \cdot p \end{cases} \quad (2.25)$$

Cette équation nous donne une caractérisation du minimum de J en fonction de la variable adjointe p .

Par conséquent, la minimisation de J passe par la résolution du système d'optimalité suivant :

$$\left\{ \begin{array}{l} \frac{dX}{dt} + F(X) + D_1 \cdot CL_1 + D_2 \cdot CL_2 = 0 \\ X(0) = 0 \\ \frac{dp}{dt} - \left[\frac{\partial F}{\partial X}\right]^t p = C^t (C \cdot X - X_{obs}) \\ p(T) = 0 \\ \nabla_{CL_1} J = D_1^t \cdot p \\ \nabla_{CL_2} J = D_2^t \cdot p \end{array} \right. \quad (2.26)$$

⁵voir annexe A

⁶voir annexe A

2.1.4 Contrôle d'un paramètre

Il s'agit ici de contrôler la modélisation elle-même de la rivière, par exemple la géométrie et sa schématisation. Mais aussi de contrôler les paramètres des formulations empiriques qui sont utilisées par la simulation, par exemple le coefficient de rugosité qui exprime le frottement de l'écoulement sur le sol et dans la végétation.

Soit le modèle général suivant :

$$\begin{cases} \frac{dX}{dt} + F(X, K) = 0 \\ X(0) = 0 \end{cases} \quad (2.27)$$

où F est un opérateur d'équations aux dérivées partielles, X est la variable d'état du système et K la variable de contrôle de notre étude. Ce système sera résolu sur l'intervalle de temps $[0, T]$

Soit X_{obs} , un jeu de données que nous considérerons continu en temps, χ l'espace des prévisions tel que $X \in \chi$ et χ_{obs} celui des observations tel que $X_{obs} \in \chi_{obs}$.

Soit l'opérateur C projetant une variable de l'espace des prévisions sur celui des observations :

$$C : \chi \rightarrow \chi_{obs}$$

Comparons maintenant mesures et observations grâce à la fonctionnelle :

$$J(K) = \frac{1}{2} \|CX(K) - X_{obs}\|_{\chi_{obs}}^2 + \frac{1}{2} \|K - K_0\|_{\mathcal{K}}^2 \quad (2.28)$$

avec $\|\cdot\|_{\chi_{obs}}^2$ la norme définie dans l'espace des observations et $\|\cdot\|_{\mathcal{K}}^2$ celle de l'espace des paramètres. K_0 est la valeur *a priori* du paramètre.

Le problème est de trouver K^* minimisant J , cette minimisation est caractérisée par la condition d'Euler :

$$\nabla_K J(K^*) = 0 \quad (2.29)$$

En pratique K^* est calculé par un algorithme d'optimisation du type gradient conjugué.

Le but de la méthode est de calculer le **gradient de J**

Soit la Gâteaux dérivée⁷ du modèle (2.27) suivant la direction δK de K :

$$\begin{cases} \frac{d\widehat{X}}{dt} + \left[\frac{\partial F(X, K)}{\partial X} \right] \cdot \widehat{X} + \left[\frac{\partial F(X, K)}{\partial K} \right] \cdot \widehat{K} = 0 \\ \widehat{X}(0) = 0 \end{cases} \quad (2.30)$$

avec \widehat{X} la dérivée directionnelle de X suivant δK et \widehat{K} la dérivée directionnelle de K suivant δK .

Soit la Gâteaux dérivée⁸ de la fonction coût (2.28)

$$(\nabla_K J, \widehat{K}) = (C.X - X_{obs}, C\widehat{X}) \quad (2.31)$$

⁷voir annexe A

⁸voir annexe A

Calculer le gradient de J revient à montrer que $\nabla_K J$ est linéaire par rapport à δK . Soit p , que nous appellerons l'adjoint, une variable de même dimension que \widehat{X} . Cette variable est multipliée au sens du produit scalaire avec le modèle (2.30).

$$\left(\frac{d\widehat{X}}{dt}, p\right) + \left(\left[\frac{\partial F(X, K)}{\partial X}\right] \cdot \widehat{X}, p\right) + \left(\left[\frac{\partial F(X, K)}{\partial K}\right] \cdot \widehat{K}, p\right) = 0 \quad (2.32)$$

Après une intégration par parties⁹ sur le premier terme de l'équation précédente, nous obtenons :

$$\left(\widehat{X}(T), p(T)\right) - \left(\frac{dp}{dt}, \widehat{X}\right) + \left(\left[\frac{\partial F(X, K)}{\partial X}\right] \cdot \widehat{X}, p\right) + \left(\left[\frac{\partial F(X, K)}{\partial K}\right] \cdot \widehat{K}, p\right) = 0 \quad (2.33)$$

avec la définition suivante,

$$(AX, Y) = (X, A^t Y) \quad (2.34)$$

l'équation (2.33) devient :

$$\left(\widehat{X}(T), p(T)\right) + \left(-\frac{dp}{dt} + \left[\frac{\partial F(X, K)}{\partial X}\right]^t p, \widehat{X}\right) + \left(\left[\frac{\partial F(X, K)}{\partial K}\right]^t p, \widehat{K}\right) = 0 \quad (2.35)$$

Soit la Gâteaux dérivée¹⁰ de J suivant δK (2.31) que nous réécrivons de manière différente :

$$\left(\widehat{K}, \nabla_K J\right) - \left(C^t (C.X - X_{obs}), \widehat{X}\right) = 0 \quad (2.36)$$

Posons p solution de

$$\begin{cases} \frac{dp}{dt} - \left[\frac{\partial F}{\partial X}\right]^t p = C^t (C.X - X_{obs}) \\ p(T) = 0 \end{cases} \quad (2.37)$$

En comparant (2.35) et (2.36), nous obtenons :

$$\nabla_K J = \left[\frac{\partial F(X, K)}{\partial K}\right]^t p \quad (2.38)$$

Cette équation nous donne une caractérisation du minimum de J en fonction de la variable adjointe p .

Par conséquent, la minimisation de J passe par la résolution du système d'optimalité suivant :

$$\begin{cases} \frac{dX}{dt} + F(X, K) = 0 \\ X(0) = 0 \\ \frac{dp}{dt} - \left[\frac{\partial F}{\partial X}\right]^t p = C^t (C.X - X_{obs}) \\ p(T) = 0 \\ \nabla_K J = \left[\frac{\partial F(X, K)}{\partial K}\right]^t p \end{cases} \quad (2.39)$$

⁹voir annexe A

¹⁰voir annexe A

2.2 Assimilation de données et Analyse de sensibilité

Lorsque l'on étudie le calage d'un modèle, il est logique de se demander quelle est l'influence de la modification de certains paramètres sur les résultats du modèle. Ainsi en modifiant une variable, va-t-on perturber de manière significative le comportement de l'écoulement ?

Par exemple, quelle-est l'incertitude sur le niveau atteint par l'inondation qui résulte de l'incertitude sur le coefficient de rugosité ? ou alors dans un système d'annonce de crue comment se répercute en aval l'incertitude sur le débit introduit en amont ?

Pour répondre à cette question légitime, une analyse de la sensibilité du modèle à la perturbation des variables d'entrée du système est alors nécessaire. De nombreux travaux ont été effectués dans le passé ; d'abord basée sur la théorie des perturbations, Wigner [21], l'analyse de sensibilité s'est ensuite orientée vers l'utilisation de l'assimilation de données, Cacuci [1].

En se donnant une fonction réponse (ou critère de sensibilité) qui est une fonction des paramètres de contrôle, le but de l'analyse de sensibilité est de calculer le gradient de cette fonction réponse par rapport aux paramètres afin d'en déduire leur sensibilité aux perturbations. Généralement, l'analyse de sensibilité se fait à partir des équations directes régissant l'évolution de l'état du système. Cette conception des choses ne tient pas compte du fait que les résultats d'un modèle ne dépendent pas seulement des vecteurs de variables tels que la condition initiale ou les conditions aux limites mais aussi de l'imperfection du modèle par rapport à la réalité représenté par les mesures (c'est-à-dire les observations). La prise en compte des observations dans l'analyse de sensibilité nécessite de considérer les équations adjointes au second ordre.

Nous verrons dans cette partie, grâce à un exemple proposé par Le Dimet [10], que les équations adjointes du premier ordre ne sont pas suffisantes pour prendre en compte tous les paramètres et par conséquent génèrent de nombreuses erreurs. Par la suite, la méthode de l'analyse de sensibilité au second ordre sera explicitée.

2.2.1 Illustration de l'importance du second ordre

Considérons le modèle suivant :

$$\begin{cases} \frac{dX}{dt} = \alpha \cdot X \\ X(0) = U \end{cases} \quad (2.40)$$

où X est la variable d'état définie sur l'intervalle de temps $[0, 1]$ et α une constante.

Soit X_{obs} une observation de X que nous supposons égale à une constante β dans l'intervalle de temps $[0, 1]$.

Soit la fonctionnelle à minimiser :

$$J(U) = \frac{1}{2} \int_0^1 (X(t) - \beta)^2 dt \quad (2.41)$$

Soit la fonction réponse :

$$G(\alpha) = \int_0^1 X(t) dt \quad (2.42)$$

Nous savons que la solution de l'équation différentielle (2.40) est :

$$X(t) = U.e^{\alpha.t} \quad (2.43)$$

Soit l'expression de J en fonction de U :

$$J(U) = \frac{1}{2} \int_0^1 (U.e^{\alpha.t} - \beta)^2 dt \quad (2.44)$$

Grâce à l'équation d'Euler caractérisant le minimum de J, nous obtenons la valeur optimale de U

$$U_{opt} = \frac{2\beta}{e^\alpha + 1} \quad (2.45)$$

Soit G en fonction de α à l'optimum U_{opt}

$$G(\alpha) = \frac{2\beta}{\alpha} \frac{e^\alpha - 1}{e^\alpha + 1} \quad (2.46)$$

La sensibilité exacte de G par rapport à α est déterminée par sa dérivée par rapport à α :

$$S = G'(\alpha) = \frac{2\beta}{\alpha(e^\alpha + 1)} \left[\frac{2e^\alpha}{e^\alpha + 1} - \frac{e^\alpha - 1}{\alpha} \right] \quad (2.47)$$

Nous allons maintenant calculer la sensibilité grâce aux équations adjointes au premier ordre.

Grâce aux développements précédents nous pouvons écrire le système adjoint du système (2.40) :

$$\begin{cases} \frac{dP}{dt} + \alpha P = 1 \\ P(1) = 0 \end{cases} \quad (2.48)$$

La variable adjointe P, solution de (2.48), s'écrit :

$$P(t) = \frac{1}{\alpha} [1 - e^\alpha . e^{-\alpha.t}] \quad (2.49)$$

Le gradient de G est alors donné par :

$$\begin{aligned} G' &= - \int_0^1 X(t).P(t)dt \\ &= - \int_0^1 \frac{2\beta}{e^\alpha + 1} e^{\alpha.t} . \frac{1}{\alpha} [1 - e^\alpha . e^{-\alpha.t}] \\ &= \frac{2\beta}{\alpha(e^\alpha + 1)} \left[e^\alpha - \frac{e^\alpha - 1}{\alpha} \right] \end{aligned} \quad (2.50)$$

Les expressions (2.47) et (2.50) sont différentes. La seconde ne tient pas compte de la dérivée de U_{opt} par rapport à α . Cette erreur est couramment commise lors de la mise en œuvre de l'analyse de sensibilité.

Pour prendre en compte les observations dans le calcul de la sensibilité, il va falloir non pas considérer le modèle direct comme modèle à partir duquel la sensibilité va être calculée, mais plutôt le système d'optimalité qui, lui, prend en compte les observations et qui surtout est le véritable modèle qui peut nous permettre de considérer l'état du système de manière optimale.

2.2.2 Analyse de sensibilité et second ordre

Reprenons le formalisme introduit précédemment dans la section sur l'assimilation de données. Supposons que le modèle généralisé a la formulation suivante :

$$\frac{d\mathcal{X}}{dt} + \mathcal{F}(\mathcal{X}, \mathcal{K}) = 0 \quad (2.51)$$

où \mathcal{X} est la variable décrivant l'état du système (éventuellement dépendant du temps) et \mathcal{K} l'une des variables de contrôle du modèle (condition initiale, conditions aux limites, paramètres empiriques ou non, ou une combinaison de ces paramètres).

Soit \mathcal{G} une fonction scalaire donnée, dépendante de \mathcal{X} . Cette fonction est appelée fonction réponse ou critère de sensibilité.

La sensibilité de \mathcal{F} par rapport à \mathcal{K} est le gradient de \mathcal{G} par rapport à \mathcal{X} .

Comme nous l'avons dit précédemment, ce n'est pas le modèle direct qui doit servir de base au raisonnement, mais bien le système d'optimalité.

Soit K un vecteur de paramètres du modèle liés par le système suivant :

$$\begin{cases} \frac{dX}{dt} = F(X, K) \\ X(0) = U \end{cases} \quad (2.52)$$

dont le système adjoint associé est :

$$\begin{cases} \frac{dP}{dt} + \left[\frac{\partial F}{\partial X} \right]^t . P = C^t (CX - X_{obs}) \\ P(T) = 0 \end{cases} \quad (2.53)$$

Afin de mener à bien l'étude de sensibilité, nous allons nous rapprocher du formalisme introduit précédemment, en posant :

$$\mathcal{X} = (X, P) \quad (2.54)$$

$$\mathcal{F}(\mathcal{X}, \mathcal{K}) = \begin{pmatrix} \frac{dX}{dt} - F(X, K) \\ \frac{dP}{dt} + \left[\frac{\partial F}{\partial X} \right]^t . P - C^t (CX - X_{obs}) \end{pmatrix} \quad (2.55)$$

Pour estimer la sensibilité par rapport au paramètre K , appliquons une perturbation dans la direction δk sur le paramètre K . Les dérivées directionnelles \widehat{X} de X et \widehat{P} de P dans la direction δk sont obtenues comme solution du système couplé :

$$\begin{cases} \frac{d\widehat{X}}{dt} = \left[\frac{\partial F}{\partial X}(X, K) \right] \widehat{X} + \left[\frac{\partial F}{\partial K}(X, K) \right] \delta k \\ \widehat{X}(0) = \widehat{U} \end{cases} \quad (2.56)$$

$$\begin{cases} \frac{d\widehat{P}}{dt} + \left[\frac{\partial^2 F}{\partial X^2}(X, K) \widehat{X} \right]^t P + \left[\frac{\partial^2 F}{\partial X \partial K}(X, K) \delta k \right]^t P + \left[\frac{\partial F}{\partial X}(X, K) \right] \widehat{P} = C^t C \widehat{X} \\ \widehat{P}(T) = 0 \\ \widehat{P}(0) = 0 \end{cases} \quad (2.57)$$

Posons Q et R les variables adjointes au second ordre.

Nous effectuons le produit scalaire de (2.56) par Q et (2.57) par R , puis nous sommes les deux expressions obtenues et nous intégrons par parties¹¹ entre les instants 0 et T .

Par conséquent :

$$\begin{aligned}
 & (\widehat{X}(T), Q(T)) - (\widehat{X}(0), Q(0)) + (\widehat{P}(T), R(T)) - (\widehat{P}(0), R(0)) \\
 & + \left(\widehat{X}, -\frac{dQ}{dt} - \left[\frac{\partial F}{\partial X} \right]^t Q + \left[\frac{\partial^2 F}{\partial X^2} R \right]^t P - C^t C R \right) \\
 & + \left(\widehat{P}, -\frac{dR}{dt} + \left[\frac{\partial F}{\partial X} \right] R \right) - \left(\delta k, \left[\frac{\partial F}{\partial K} \right]^t Q + \left[\frac{\partial^2 F}{\partial X \partial K} R \right]^t .P \right) = 0
 \end{aligned} \tag{2.58}$$

Considérons une fonction réponse sous la forme intégrale suivante :

$$\mathcal{G} = \int_0^T G(X) dt \tag{2.59}$$

où G est une fonction scalaire.

Posons Q et R solution du système suivant :

$$\left\{ \begin{array}{l} \frac{dQ}{dt} + \left[\frac{\partial F}{\partial X} \right]^t Q - \left[\frac{\partial^2 F}{\partial X^2} R \right]^t P + C^t C R = \left[\frac{\partial G}{\partial X} \right] \\ \frac{dR}{dt} - \left[\frac{\partial F}{\partial X} \right] R = 0 \\ Q(T) = 0 \\ Q(0) = 0 \end{array} \right. \tag{2.60}$$

Le gradient de la fonction réponse est donné par :

$$\nabla \mathcal{G} = \int_0^T \left[\frac{\partial F}{\partial K} \right]^t .Q - \left[\frac{\partial^2 F}{\partial X \partial K} R \right]^t .P dt \tag{2.61}$$

Nous avons travaillé de façon à ce que le modèle généralisé soit associé au système d'optimalité. Ainsi, les observations font partie intégrante de ce modèle généralisé. Il est maintenant logique de se demander quelle est l'influence de ces observations sur le modèle, c'est-à-dire quelles sont les observations nécessaires à l'amélioration significative de la prévision. Dans ce cas, nous allons reprendre le formalisme précédent et poser $\mathcal{K} = X_{obs}$.

Soit une fonction réponse \mathcal{G} fonction de X . Une démarche identique à ce que nous avons vu précédemment nous permet d'exprimer la sensibilité de la fonction réponse par rapport aux observations :

$$\nabla \mathcal{G} = -C.R = -\int_0^T C.R(t) dt \tag{2.62}$$

2.2.3 Algorithme de calcul de la sensibilité

Afin de calculer la sensibilité, nous devons résoudre le système (2.60).

Nous allons donc maintenant proposer un algorithme simple pour résoudre ce problème :

¹¹voir annexe A

a) Résoudre le système suivant en choisissant une condition initiale pour R

$$\begin{cases} \frac{dR}{dt} - \left[\frac{\partial F}{\partial X} \right] R = 0 \\ R(0) = v \end{cases} \quad (2.63)$$

Remarque : $\left[\frac{\partial F}{\partial X} \right]$ est le Jacobien de F . Ce modèle, dit tangent, sera calculé automatiquement grâce à un logiciel de différentiation automatique dont nous discuterons plus tard.

b) Intégrer de manière rétrograde le reste du système (2.60) :

$$\begin{cases} \frac{dQ}{dt} + \left[\frac{\partial F}{\partial X} \right]^t Q = \left[\frac{\partial^2 F}{\partial X^2} R \right]^t P - C^t C R + \left[\frac{\partial G}{\partial X} \right] \\ Q(T) = 0 \end{cases} \quad (2.64)$$

Nous obtenons alors $Q(0)$ comme une fonction de v

$$Q(0) = Q(0, v) \quad (2.65)$$

Nous devons calculer une valeur optimale, v^* , de sorte que

$$Q(0) = Q(0, v^*) = 0 \quad (2.66)$$

Calcul du produit Hessien \times Vecteur

Considérons le système d'optimalité introduit précédemment :

$$\begin{cases} \frac{dX}{dt} = F(X, K) & , t \in [0, T] \\ X(0) = U \\ \frac{dP}{dt} + \left[\frac{\partial F}{\partial X} \right]^t . P = C^t (CX - X_{obs}) & , t \in [0, T] \\ P(T) = 0 \\ \nabla_K J = - \left[\frac{\partial F}{\partial K} \right]^t . P = 0 \\ \nabla_U J = -P(0) = 0 \end{cases} \quad (2.67)$$

En appliquant une perturbation $h = (h_U, h_K)$ sur les paramètres K et U dans l'équation (2.67), nous obtenons le système suivant :

$$\begin{cases} \frac{d\widehat{X}}{dt} = \left[\frac{\partial F}{\partial X} \right] \widehat{X} + \left[\frac{\partial F}{\partial K} \right] h_K \\ \widehat{X}(0) = h_U \end{cases} \quad (2.68)$$

où \widehat{X} est la perturbation sur X .

Nous allons appliquer les mêmes perturbations sur le système (2.67).

$$\begin{cases} \frac{d\widehat{P}}{dt} + \left[\frac{\partial^2 F}{\partial X^2} \widehat{X} + \frac{\partial^2 F}{\partial X \partial K} h_k \right]^t . P + \left[\frac{\partial F}{\partial X} \right]^t \widehat{P} = C^t C \widehat{X} & , t \in [0, T] \\ \widehat{P}(T) = 0 \end{cases} \quad (2.69)$$

$$\widehat{\nabla_u J} = -\widehat{P}(0) = 0 \quad (2.70)$$

$$\widehat{\nabla_K J} = - \left[\frac{\partial^2 F}{\partial X \partial K} \widehat{X} + \frac{\partial^2 F}{\partial K^2} h_k \right]^t . P - \left[\frac{\partial F}{\partial X} \right]^t \widehat{P} = 0 \quad (2.71)$$

où \widehat{P} est la perturbation de P .

Le Hessien est obtenu en exhibant la dépendance linéaire de $\widehat{\nabla_K J}$ et $\widehat{\nabla_u J}$ par rapport à h .

Soit les variables adjointes au second ordre Q et R qui seront définies explicitement par la suite.

Nous allons multiplier (au sens du produit scalaire) l'équation (2.68) par Q et l'équation (2.69) par R :

$$\left(Q, \frac{d\widehat{X}}{dt} \right) = \left(Q, \left[\frac{\partial F}{\partial X} \right] \widehat{X} + \left[\frac{\partial F}{\partial K} \right] h_k \right) \quad (2.72)$$

$$\left(R, \frac{d\widehat{P}}{dt} + \left[\frac{\partial^2 F}{\partial X^2} \widehat{X} + \frac{\partial^2 F}{\partial X \partial K} h_k \right]^t . P + \left[\frac{\partial F}{\partial X} \right]^t \widehat{P} \right) = (R, C^t C \widehat{X}) \quad (2.73)$$

Puis sommons ces deux équations :

$$\begin{aligned} & \left(Q, \frac{d\widehat{X}}{dt} \right) + \left(R, \frac{d\widehat{P}}{dt} \right) = \left(Q, \left[\frac{\partial F}{\partial X} \right] \widehat{X} + \left[\frac{\partial F}{\partial K} \right] h_k \right) \\ & - \left(R, \left[\frac{\partial^2 F}{\partial X^2} \widehat{X} + \frac{\partial^2 F}{\partial X \partial K} h_k \right]^t . P - \left[\frac{\partial F}{\partial X} \right]^t \widehat{P} \right) + (R, C^t C \widehat{X}) \end{aligned} \quad (2.74)$$

Après intégration par partie¹² du membre de gauche et en utilisant la propriété du produit scalaire par rapport à la transposition :

$$(Ax, y) = (x, A^t y) \quad (2.75)$$

on obtient :

$$(\widehat{X}(T), Q(T)) - (\widehat{X}(0), Q(0)) + (\widehat{P}(T), R(T)) - (\widehat{P}(0), R(0)) = \quad (2.76)$$

$$\left(\widehat{X}, \frac{dQ}{dt} + \left[\frac{\partial F}{\partial X} \right]^t Q - \left[\frac{\partial^2 F}{\partial X^2} R \right]^t P + C^t C R \right) \quad (2.77)$$

$$+ \left(\widehat{P}, \frac{dR}{dt} - \left[\frac{\partial F}{\partial X} \right] R \right) \quad (2.78)$$

$$+ \left(h_k, \left[\frac{\partial F}{\partial K} \right]^t Q - \left[\frac{\partial^2 F}{\partial X \partial K} R \right]^t . P \right) \quad (2.79)$$

¹²voir annexe A

Définissons Q et R comme solutions du système adjoint au second ordre (ASO) introduit par Le Dimet et al [9] :

$$\begin{cases} \frac{dQ}{dt} + \left[\frac{\partial F}{\partial X} \right]^t Q - \left[\frac{\partial^2 F}{\partial X^2} R \right]^t P + C^t C R = 0 \\ \frac{dR}{dt} - \left[\frac{\partial F}{\partial X} \right] R = 0 \\ Q(T) = 0 \end{cases} \quad (2.80)$$

Par conséquent, il reste

$$- \left(\widehat{X}(0), Q(0) \right) - \left(\widehat{P}(0), R(0) \right) + \left(h_k, \left[\frac{\partial F}{\partial K} \right]^t Q - \left[\frac{\partial^2 F}{\partial X \partial K} R \right]^t . P \right) = 0 \quad (2.81)$$

Soit H , la matrice Hessienne de la fonction coût J :

$$H = \begin{pmatrix} H_{uu} & H_{uk} \\ H_{ku} & H_{kk} \end{pmatrix}$$

Par définition, nous avons

$$\nabla \widehat{J}(\widehat{U}, v) = H(U).v = -\widehat{P}(0)$$

avec $\nabla \widehat{J}(\widehat{U}, v)$ la perturbation induite sur $\nabla J(U)$ par v .

Posons $R(0) = v$.

Nous obtenons alors :

$$H(U).v = -P(0) \quad (2.82)$$

Comment trouver v^* tels que $Q(0, v^*) = 0$?

Posons (Q_1, R) solution de :

$$\begin{cases} \frac{dQ_1}{dt} + \left[\frac{\partial F}{\partial X} \right]^t Q_1 - \left[\frac{\partial^2 F}{\partial X^2} R \right]^t P = -C^t C R \\ \frac{dR}{dt} - \left[\frac{\partial F}{\partial X} \right] R = 0 \\ Q_1(T) = 0 \\ R(0) = v \end{cases} \quad (2.83)$$

et Q_2 solution de :

$$\begin{cases} \frac{dQ_2}{dt} + \left[\frac{\partial F}{\partial X} \right]^t Q_2 = \left[\frac{\partial G}{\partial X} \right] \\ Q_2(T) = 0 \end{cases} \quad (2.84)$$

En utilisant la linéarité de la première équation de (2.60), nous pouvons écrire que

$$Q = Q_1 + Q_2 \quad (2.85)$$

Par conséquent

$$Q(0) = Q_1(0) + Q_2(0) \quad (2.86)$$

Nous remarquons que le système (2.83) est équivalent au système (2.80) introduit précédemment, il vient donc

$$Q_1(0) = H.v \quad (2.87)$$

Par suite nous avons

$$Q(0) = H.v + Q_2(0) \quad (2.88)$$

H étant symétrique définie positive¹³, il existe donc un unique v^* tel que $Q(0) = 0$. Afin de trouver ce v^* , il suffit de le caractériser par

$$f(v^*) = \min f(v) \quad (2.89)$$

où f est définie par

$$f(x) = \frac{1}{2} (Hx, x) + (Q_2(0), x) \quad (2.90)$$

La minimisation de cette fonction par un algorithme de descente nous permettra d'obtenir une valeur optimale de v afin de résoudre le système (2.63) et ainsi d'initialiser le calcul de la sensibilité.

2.3 Conclusion

Nous avons présenté dans ce chapitre la théorie de l'assimilation de données variationnelle largement utilisée en météorologie et en océanographie. Le but de ce travail de thèse sera d'appliquer ces méthodes aux équations de l'hydraulique fluviale, afin notamment de tester l'estimation de certains paramètres. Actuellement, la recherche de l'état du modèle en hydraulique (c'est à dire l'estimation de paramètres) se fait de manière plus ou moins manuelle grâce à une succession d'exécution du code direct. Nous avons vu que, grâce à des observations faites sur le domaine étudié, il est possible d'estimer un jeu de paramètres optimal afin d'améliorer la prévision.

L'estimation de paramètres nécessite la connaissance de la sensibilité du modèle à la perturbation des variables de contrôle. Cette analyse doit être menée grâce au développement du code adjoint au second ordre, qui permet de combiner tout les types d'informations, modèles et observations, ceci afin d'estimer au mieux le comportement des variables d'état quelles que soient les modifications des paramètres du modèle. Généralement, l'étude de sensibilité est faite à partir des méthodes de Monté Carlo basé sur une succession d'exécution du code direct. L'assimilation de données permet d'effectuer une analyse de sensibilité avec une seule exécution du code adjoint.

La puissance du code adjoint réside dans le fait qu'un même code va permettre de caler le modèle, de faire l'analyse de sensibilité et d'étudier les erreurs du modèle. De plus, si ce code a été dérivé par rapport à tous les paramètres possibles, un seul code sera suffisant pour tout faire. C'est-à-dire qu'une fois passée l'étape difficile de l'écriture du code adjoint, l'outil est alors écrit pour envisager toutes les études citées.

¹³voir annexe A

Chapitre 3

Les modèles d'hydraulique

Il existe différents types de systèmes de modélisation afin de mener à bien une étude hydraulique. La modélisation monodimensionnelle va intervenir dans les cas de grand domaine où seul le comportement général de l'écoulement est recherché. La modélisation 2D sera de préférence choisie pour les domaines de petite taille ainsi que pour les études demandant plus de précisions que le seul comportement général de l'écoulement. Toutefois, l'étude d'un grand domaine peut nécessiter une connaissance précise de l'évolution de l'inondation sur une petite partie de celui-ci. Dans ce cas, on envisagera de préférence un couplage entre un modèle numérique 1D, modélisant la majeure partie du domaine, et un modèle numérique 2D, sur la partie nécessitant une précision supérieure.

Une fois un système choisi, on ne fait pas le même modèle selon les objectifs :

- Quel est le but de l'étude hydraulique ? On ne construira pas identiquement un modèle suivant que l'on veuille étudier le comportement de l'écoulement près de la pile d'un pont ou dans une vallée gigantesque.
- Quelle est la taille du domaine ? En effet, suivant que l'on va travailler sur un petit domaine, un bassin versant par exemple, ou un domaine de grande taille comme une rivière entière, nous n'allons pas utiliser le même type de modèle numérique pour des raisons de temps de calcul ou de capacité machine.

Dans ce travail de thèse nous avons choisi d'étudier le modèle Carima résolvant les équations de Saint Venant 1D et le modèle LMCFLD basé sur la résolution de ces mêmes équations en dimension 2. Ces deux modèles numériques seront présentés dans ce chapitre.

3.1 Un modèle d'hydraulique fluviale monodimensionnel : Carima

3.1.1 Présentation du logiciel Carima

Le système Carima (CAlcul de Rivières MAillées) est un outil de calcul des écoulements à surface libre des régions fluviales (rivières et zones inondables) et des réseaux d'irrigation. Il inclut également des modules spécifiques pour la modélisation d'organes de régulation et de qualité des eaux.

La description des cours d'eau et des plaines inondables environnantes est basée sur une représentation particulière des domaines d'écoulement et de l'écoulement lui-même entre ces différents domaines.

3.1. UN MODÈLE D'HYDRAULIQUE FLUVIALE MONODIMENSIONNEL : CARIMA

1. L'écoulement du ou des lits mineurs de la rivière est considéré comme unidimensionnel et calculé par les équations de Saint Venant (cf (3.1) et (3.2)). Chaque point de calcul correspond à un profil en travers, dont la géométrie est définie à partir de couples (largeurs, hauteurs) par l'utilisateur.

Ces points sont reliés par des tronçons de calcul unidimensionnels (voir la figure 3.1). Ces tronçons peuvent constituer des réseaux maillés permettant de simuler toutes les particularités de la rivière (affluent, île, etc...).

L'état hydraulique est défini par le niveau de la surface libre et le débit à chaque point de calcul.

Remarque : La difficulté d'une modélisation monodimensionnelle du lit de la rivière réside dans le choix des sections de calcul. Les coupes transversales mesurées par le géomètre doivent rendre compte de toutes les particularités physiques du cours d'eau. La courbure du lit par exemple va apparaître dans la définition des sections à travers un coefficient de rugosité plus ou moins élevé suivant que l'on se trouve à l'intérieur du virage ou à l'extérieur.

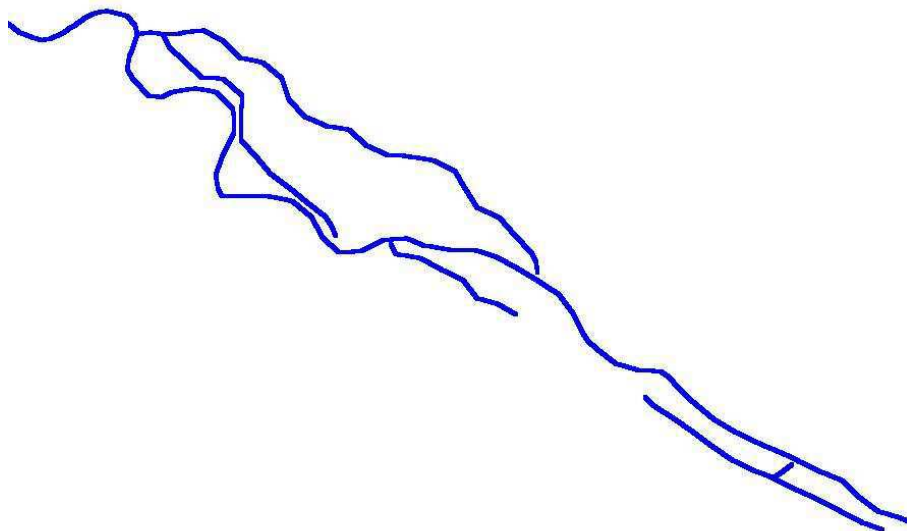


FIG. 3.1 – Visualisation de la modélisation de la Loire dans le Val d'Ouzouer par le logiciel Carima.

2. L'écoulement dans la plaine inondable est représenté par une modélisation dite à casiers (ou quasi-2D) et des équations simplifiées de l'écoulement (sans termes d'inertie). L'écoulement bidimensionnel dans la plaine inondable est simulé par une subdivision de la zone par un jeu de casiers communiquant entre eux (voir la figure 3.3 et le dessin 3.2). Seul le volume de stockage de chaque casier permet de considérer le comportement de l'eau dans la plaine inondable. Le découpage en casiers de la plaine inondable n'est pas simple. Il doit être assez précis pour permettre de visualiser au mieux les directions privilégiées de l'écoulement tout en schématisant le domaine. Ainsi, les casiers ne sont pas déterminés seulement en fonction des frontières naturelles du terrain (routes, colline, etc ...) mais aussi en fonction de considérations plus hydrauliques ou mêmes économiques (lorsque le but de l'étude est l'impact économique d'une inondation).

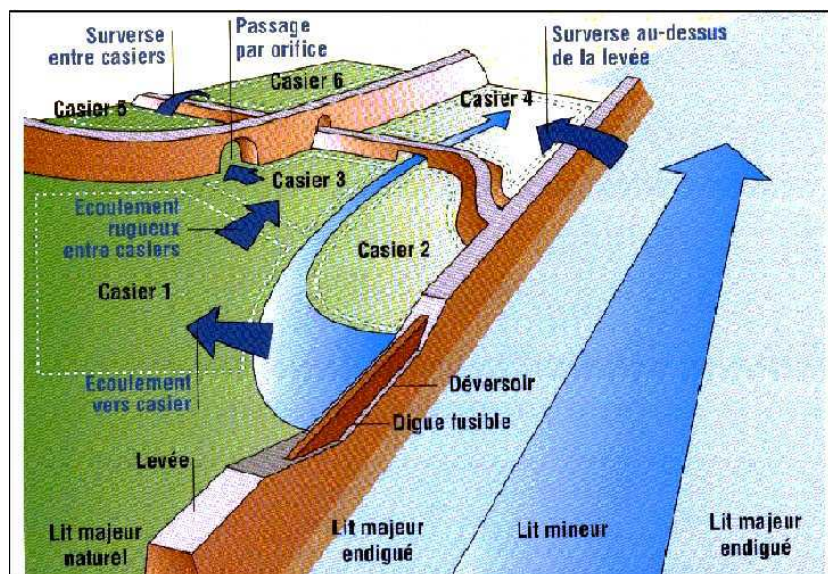


FIG. 3.2 – Représentation schématique du découpage d'une plaine inondable en casier.

Cette représentation schématique "en casiers" reproduit assez fidèlement le comportement d'une plaine d'inondation où les échanges avec la rivière et entre les différentes zones du lit majeur sont essentiellement contrôlés par des déversements ou des ouvrages. Elle est cependant couramment utilisée pour reproduire tous types d'écoulements dans le champ d'inondation. L'appellation "casier" est alors abusive (parce qu'elle suppose implicitement une surface où le niveau d'eau est uniforme) mais la procédure de calcul reste valide (le casier est alors un point du champ d'inondation qui échange avec d'autres points du lit majeur, et auquel on affecte une capacité de stockage équivalente à celle de son environnement).

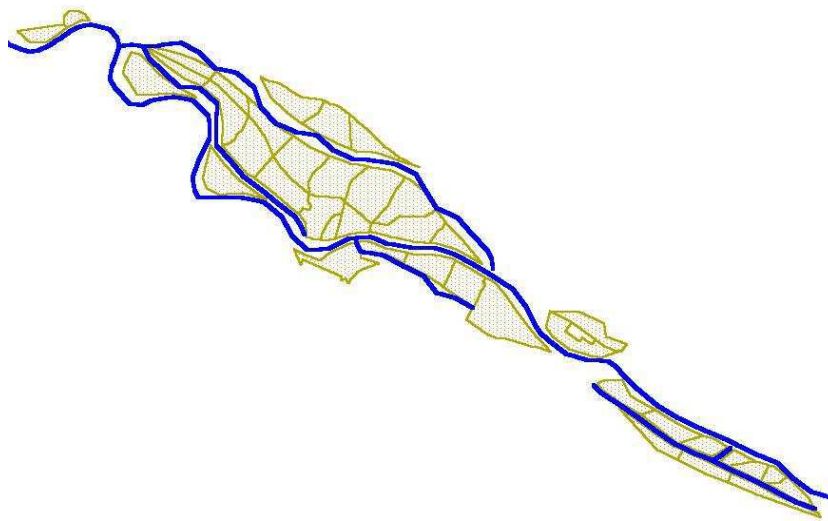


FIG. 3.3 – Visualisation de la modélisation de la plaine inondable couplée à celle de la Loire dans le Val d'Ouzouer par le logiciel Carima.

3.1. UN MODÈLE D'HYDRAULIQUE FLUVIALE MONODIMENSIONNEL : CARIMA

Le calcul de l'écoulement bidimensionnel est basé sur la continuité du volume pour chaque casier et sur les lois d'écoulement sans inertie entre les casiers.

Tout le travail développé ci-après est décrit de manière plus détaillée dans le livre de Cunge [2].

3.1.2 Les équations de Saint Venant

Les équations résolues dans Carima sont les équations classiques de l'écoulement unidimensionnel à surface libre de Saint Venant décrites dans le livre de Cunge [2]. Ces équations seront utilisées pour tout tronçon fluvial qui joint deux points de calcul successifs et discrétisées en fonction des conditions hydrauliques et des caractéristiques de ces deux points.

La résolution des équations de Saint Venant donne l'évolution de la cote du plan d'eau $z(x, t)$ et du débit $Q(x, t)$ en fonction de la distance longitudinale x et du temps t .

Les équations de Saint Venant se décomposent comme suit :

- l'équation de continuité exprime la conservation de masse du fluide

$$\frac{1}{b} \cdot \frac{\partial Q}{\partial x} + \frac{\partial z}{\partial t} = 0 \quad (3.1)$$

- l'équation dynamique exprime l'équilibre entre les forces motrices (pente et inertie) et les forces résistantes de frottement

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x} \left(\frac{\alpha Q^2}{A} \right) + g \cdot A \cdot \frac{\partial z}{\partial x} + g \cdot A \cdot S_f = 0 \quad (3.2)$$

où $b(z)$ et $A(z)$ sont respectivement la largeur du plan d'eau et l'aire de la section mouillée. Ce sont donc des fonctions de la géométrie de la section d'écoulement. g est l'accélération de la gravité. α est le coefficient de Boussinesq.

Le terme S_f est le gradient longitudinal d'énergie qui résulte des forces résistantes de frottement. Sa signification et son expression sont précisés ci-après.

La pente de frottement S_f

En écoulement graduellement varié (c'est-à-dire dans le domaine de validité des équations de Saint Venant), les ingénieurs utilisent des relations empiriques pour l'expression de la dissipation d'énergie. La formule de Manning-Strickler est la plus populaire en hydraulique fluviale et dans les logiciels de simulation.

$$S_f = \frac{Q|Q|}{K^2}$$

K la débitance, exprime la qualité intrinsèque de la section à laisser couler l'eau. Cette variable dépend donc de la géométrie et des propriétés de frottement entre la veine fluide et les parois.

La débitance K

La formule de Strickler fait intervenir la débitance K de la section hydraulique :

$$K = K_{str} \cdot A \cdot R_h^{\frac{2}{3}} \quad (3.3)$$

où

- $A(z)$ est l'aire de la section mouillée,
- $R_h(z)$ le rayon hydraulique,
- K_{str} est le coefficient de rugosité de Strickler.

3.1.3 Discrétisation des équations de Saint Venant

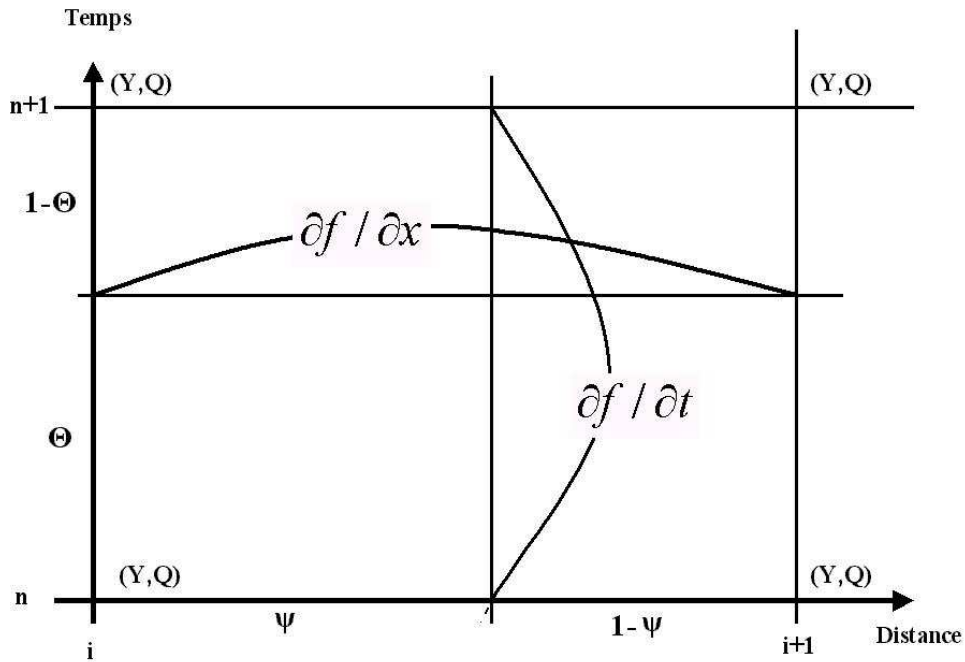


FIG. 3.4 – Schéma implicite de Preissmann utilisé dans le logiciel Carima.

Pour discrétiser les équations de Saint Venant, le logiciel Carima utilise une méthode de différences finies basée sur le schéma implicite de Preissmann (voir l'article de Preissmann [18]), défini par les règles suivantes (voir figure 3.4) :

- toute fonction f dans les équations différentielles de Saint Venant est discrétisée comme suit :

$$f = \theta \left[\Psi f_i^{n+1} + (1 - \Psi) f_{i+1}^{n+1} \right] + (1 - \theta) \left[\Psi f_i^n + (1 - \Psi) f_{i+1}^n \right] \quad (3.4)$$

- les dérivées par rapport à x s'écrivent :

$$\frac{\partial f}{\partial x} = \frac{1}{\Delta x} \left[\theta \left(f_{i+1}^{n+1} - f_i^{n+1} \right) + (1 - \theta) \left(f_{i+1}^n - f_i^n \right) \right] \quad (3.5)$$

- les dérivées temporelles s'écrivent :

$$\frac{\partial f}{\partial t} = \frac{1}{\Delta t} \left[\Psi \left(f_i^{n+1} - f_i^n \right) + (1 - \Psi) \left(f_{i+1}^{n+1} - f_{i+1}^n \right) \right] \quad (3.6)$$

3.1. UN MODÈLE D'HYDRAULIQUE FLUVIALE MONODIMENSIONNEL : CARIMA

La discrétisation donne un poids égal aux deux points de calcul successifs i et $i + 1$ aux extrémités du tronçon fluvial ($\Psi = 0, 5$). Dans la suite de ce travail nous poserons donc $\Psi = 0, 5$.

Le choix du coefficient θ se fera en par rapport à la stabilité du modèle dont nous parlerons dans la section 3.1.7.

Discrétisation de l'équation de continuité

La discrétisation de l'équation de continuité donne :

$$\frac{1}{2\Delta t} (b_{i+1}\Delta z_{i+1} - b_i\Delta z_i) + \frac{\theta}{\Delta x} (\Delta Q_{i+1} - \Delta Q_i) + \frac{1}{\Delta x} (Q_{i+1}^n - Q_i^n) = 0 \quad (3.7)$$

Si nous posons $\Delta Q_i = Q_i^{n+1} - Q_i^n$ et $\Delta z_i = z_i^{n+1} - z_i^n$, l'équation (3.7) peut se réécrire :

$$A\Delta z_{i+1} + B\Delta Q_{i+1} = C\Delta z_i + D\Delta Q_i + G \quad (3.8)$$

avec

$$\left\{ \begin{array}{l} A = \frac{b_{i+1}}{2\Delta t} \\ B = D = \frac{\theta}{\Delta x} \\ C = -\frac{b_i}{2\Delta t} \\ G = -\frac{1}{\Delta x} (Q_{i+1}^n - Q_i^n) \end{array} \right.$$

Discrétisation de l'équation dynamique

Nous utilisons l'équation (3.2) dans une forme différente. Considérons le deuxième terme :

$$\frac{\partial}{\partial x} \left(\frac{\alpha Q^2}{A} \right) = -\frac{\alpha Q^2}{A^2} \frac{\partial A}{\partial x} + \frac{2\alpha Q}{A} \frac{\partial Q}{\partial x} \quad (3.9)$$

Il vient donc, en considérant l'équation (3.1) :

$$\frac{\partial}{\partial x} \left(\frac{\alpha Q^2}{A} \right) = -\frac{\alpha Q^2}{A^2} \frac{\partial A}{\partial x} + \frac{\alpha Q}{A} \frac{\partial Q}{\partial x} - \alpha \frac{Qb}{A} \frac{\partial z}{\partial t} \quad (3.10)$$

Nous avons donc :

$$\begin{array}{cccccc} \frac{\partial Q}{\partial t} & - & \alpha \frac{Qb}{A} \cdot \frac{\partial z}{\partial t} & + & \alpha \frac{Q}{A} \cdot \frac{\partial Q}{\partial x} & - & \alpha \frac{Q^2}{A^2} \cdot \frac{\partial A}{\partial x} & + & g \cdot A \cdot \frac{\partial z}{\partial x} & + & g \cdot A \cdot S_f & = & 0 \\ I & & II & & III & & IV & & V & & VI & & \end{array}$$

Nous allons maintenant résumer la discrétisation de chacun des termes :

I Nous obtenons directement :

$$\frac{\partial Q}{\partial t} \simeq \frac{1}{2\Delta t} (\Delta Q_{i+1} + \Delta Q_i) \quad (3.11)$$

II En écrivant $\Delta A = b\Delta z$ et en négligeant les produits $\Delta Q\Delta b$, nous obtenons :

$$-\alpha \frac{Q \cdot b}{A} \frac{\partial z}{\partial t} \simeq -\frac{\alpha}{4\Delta t} (\Delta z_{i+1} + \Delta z_i) \left(\frac{Q_{i+1}^n \cdot b_{i+1}^n}{A_{i+1}^n} + \frac{Q_i^n \cdot b_i^n}{A_i^n} \right) \quad (3.12)$$

III En négligeant les produits $\Delta Q\Delta A$, nous obtenons :

$$\begin{aligned} \alpha \frac{Q}{A} \frac{\partial Q}{\partial x} &\simeq \\ &\frac{\alpha}{2\Delta x} (Q_{i+1}^n - Q_i^n) \cdot \left[\left(\frac{\theta}{(A_{i+1}^n)^2} \right) \cdot (A_{i+1}^n \Delta Q_{i+1} - Q_{i+1}^n b_{i+1}^n \Delta z_{i+1}) \right] \\ &+ \frac{\alpha}{2\Delta x} (Q_{i+1}^n - Q_i^n) \cdot \left[\left(\frac{\theta}{(A_i^n)^2} \right) (A_i^n \Delta Q_i - Q_i^n b_i^n \Delta z_i) + \frac{Q_{i+1}^n}{A_{i+1}^n} + \frac{Q_i^n}{A_i^n} \right] \\ &+ \frac{\alpha \cdot \theta}{2\Delta x} (\Delta Q_{i+1} - \Delta Q_i) \left(\frac{Q_{i+1}^n}{A_{i+1}^n} + \frac{Q_i^n}{A_i^n} \right) \end{aligned} \quad (3.13)$$

IV En négligeant les produits $\Delta Q\Delta A$, nous obtenons :

$$\begin{aligned} -\alpha \frac{Q^2}{A^2} \frac{\partial A}{\partial x} &\simeq -\Delta z_{i+1} \frac{\alpha \theta b_{i+1}^n}{4\Delta x} (v_i^n + v_{i+1}^n) \left[(v_i^n + v_{i+1}^n) - \frac{2Q_{i+1}^n (A_{i+1}^n - A_i^n)}{(A_{i+1}^n)^2} \right] \\ &+ \Delta z_i \frac{\alpha \theta b_i^n}{4\Delta x} (v_i^n + v_{i+1}^n) \left[(v_i^n + v_{i+1}^n) + \frac{2Q_i^n (A_{i+1}^n - A_i^n)}{(A_{i+1}^n)^2} \right] \\ &- \Delta Q_{i+1} \frac{\alpha \theta}{2\Delta x} (v_i^n + v_{i+1}^n) \left(\frac{A_{i+1}^n - A_i^n}{A_{i+1}^n} \right) \\ &- \Delta Q_i \frac{\alpha \theta}{2\Delta x} (v_i^n + v_{i+1}^n) \left(\frac{A_{i+1}^n - A_i^n}{A_i^n} \right) \\ &- \frac{\alpha}{4\Delta x} (v_i^n + v_{i+1}^n)^2 (A_{i+1}^n - A_i^n) \end{aligned} \quad (3.14)$$

$$\text{où } v = \frac{Q}{A}$$

V En négligeant les produits $\Delta z\Delta A$, nous obtenons directement :

$$\begin{aligned} g \cdot A \cdot \frac{\partial z}{\partial x} &\simeq \frac{g}{2\Delta x} \theta (z_{i+1}^n - z_i^n) (b_{i+1}^n \Delta z_{i+1} + b_i^n \Delta z_i) \\ &+ \frac{g}{2\Delta x} \theta (A_{i+1}^n + A_i^n) (\Delta z_{i+1} - \Delta z_i) \\ &+ \frac{g}{2\Delta x} (A_{i+1}^n + A_i^n) (z_{i+1}^n - z_i^n) \end{aligned} \quad (3.15)$$

VI En considérant l'équation (3.17), nous obtenons :

$$\begin{aligned} g \cdot A \cdot S_f &\simeq \Delta z_{i+1} \frac{g\theta}{2} \left[b_{i+1} S F 2 - 2 (A_i + A_{i+1}) Q_{i+1} |Q_{i+1}| \frac{K'_{i+1}}{2 \cdot K_{i+1}^3} \right] \\ &+ \Delta z_i \frac{g\theta}{2} \left[b_i S F 2 - 2 (A_i + A_{i+1}) Q_i |Q_i| \frac{K'_i}{2 \cdot K_i^3} \right] \\ &+ \Delta Q_{i+1} g \theta \frac{1}{2} (A_i + A_{i+1}) \frac{|Q_{i+1}|}{K_{i+1}^2} \\ &+ \Delta Q_i g \theta \frac{1}{2} (A_i + A_{i+1}) \frac{|Q_i|}{K_i^2} \\ &+ \frac{g}{2} (A_i + A_{i+1}) S F 2 \end{aligned} \quad (3.16)$$

$$\text{où } \begin{cases} K' &= \frac{\partial K}{\partial z} \\ SF2 &= \frac{1}{2} \frac{Q_{i+1}|Q_{i+1}|}{K_{i+1}^2} + \frac{1}{2} \frac{Q_i|Q_i|}{K_i^2} \end{cases}$$

Ici le coefficient S_f , c'est-à-dire la perte de charge due aux frottements sur le fond et sur les berges, dans un tronçon fluvial, est calculé comme la moyenne pondérée entre les points de calcul aux extrémités de ce tronçon fluvial. Nous rappelons ici que les points de calculs ont été tous définis par une coupe transversale par l'utilisateur.

Soit AM le point amont et AV le point aval du tronçon $AM \rightarrow AV$. La pente de frottement est alors calculée comme suit :

$$\begin{aligned} S_f &= \Psi \cdot S_{f,AM} + (1 - \Psi) \cdot S_{f,AV} \\ &= \Psi \cdot \frac{Q_{AM}|Q_{AM}|}{K_{AM}^2} + (1 - \Psi) \cdot \frac{Q_{AV}|Q_{AV}|}{K_{AV}^2} \end{aligned} \tag{3.17}$$

avec

- Ψ le coefficient de pondération spatiale ($0 \leq \Psi \leq 1$),
- $S_{f,X}$ la pente de frottement au point X du tronçon, où $X = \{AM, AV\}$,
- Q_X le débit de la section X , où $X \in \{AM, AV\}$,
- K_X la débitance de la section X , où $X = \{AM, AV\}$.

Comme nous l'avons vu précédemment, nous prenons $\Psi = 0.5$.

Tout comme nous l'avons fait pour l'équation de continuité (3.7), nous pouvons réécrire la discrétisation de l'équation (3.2) sous la forme :

$$A' \Delta z_{i+1} + B' \Delta Q_{i+1} = C' \Delta z_i + D' \Delta Q_i + G' \tag{3.18}$$

Si nous posons $\Sigma X_i = X_{i+1}^n + X_i^n$ et $\Lambda X_i = X_{i+1}^n - X_i^n$ avec $X \in \{Q, z, A, v\}$, alors les coefficients A', B', C', D' et G' s'écrivent :

$$\left\{ \begin{array}{l} A' = \frac{\alpha \theta b_{i+1}^n \cdot \Sigma v_i}{4 \Delta x} \left[\frac{2 Q_{i+1}^n \cdot \Lambda A_i}{(A_{i+1}^n)^2} - \Sigma v_i \right] - \frac{1}{4 \Delta t} \left(\frac{Q_{i+1}^n \cdot b_{i+1}^n}{A_{i+1}^n} + \frac{Q_i^n \cdot b_i^n}{A_i^n} \right) \\ \quad - \frac{\alpha}{2 \Delta x} \cdot \Lambda Q_i \cdot \frac{\theta}{(A_{i+1}^n)^2} \cdot Q_{i+1}^n b_{i+1}^n + \frac{g \theta}{2 \Delta x} [b_{i+1}^n \cdot \Lambda z_i + \Sigma A_i] \\ \quad + \frac{g \theta}{2} \left[b_{i+1} S F 2 - 2 \cdot \Sigma A_i \cdot Q_{i+1} |Q_{i+1}| \frac{K'_{i+1}}{2 \cdot K_{i+1}^3} \right] \\ B' = \frac{1}{2 \Delta t} + \frac{\alpha}{2 \Delta x} \cdot \Lambda Q_i \cdot \frac{\theta}{(A_{i+1}^n)^2} \cdot A_{i+1}^n + \frac{\alpha \cdot \theta}{2 \Delta x} \left(\frac{Q_{i+1}^n}{A_{i+1}^n} + \frac{Q_i^n}{A_i^n} \right) \\ \quad + \frac{-\alpha \theta}{2 \Delta x} \cdot \Sigma v_i \cdot \frac{\Lambda A_i}{A_{i+1}^n} + g \theta \frac{1}{2} \cdot \Sigma A_i \cdot \frac{|Q_{i+1}|}{K_{i+1}^2} \\ C' = \frac{1}{4 \Delta t} \left(\frac{Q_{i+1}^n \cdot b_{i+1}^n}{A_{i+1}^n} + \frac{Q_i^n \cdot b_i^n}{A_i^n} \right) - \frac{\alpha}{2 \Delta x} \cdot \Lambda Q_i \cdot \left[-\frac{\theta}{(A_i^n)^2} \cdot Q_i^n b_i^n + \frac{Q_{i+1}^n}{A_{i+1}^n} + \frac{Q_i^n}{A_i^n} \right] \\ \quad - \frac{\alpha \theta b_i}{4 \Delta x} \cdot \Sigma v_i \cdot \left[\Sigma v_i + \frac{2 Q_i^n \cdot \Lambda A_i}{(A_{i+1}^n)^2} \right] - \frac{g}{2 \Delta x} \cdot \theta \cdot \Lambda z_i \cdot b_i^n + \frac{g}{2 \Delta x} \cdot \theta \cdot \Sigma A_i \\ \quad - \frac{g \theta}{2} \left[b_i S F 2 - 2 \cdot \Sigma A_i \cdot Q_i |Q_i| \frac{K'_i}{2 \cdot K_i^3} \right] \\ D' = -\frac{1}{2 \Delta t} - g \theta \frac{1}{2} \Sigma A_i \frac{|Q_i|}{K_i^2} - \frac{-\alpha \theta}{2 \Delta x} \cdot \Sigma v_i \cdot \frac{\Lambda A_i}{A_i^n} - g \theta \frac{1}{2} \cdot \Sigma A_i \frac{|Q_i|}{K_i^2} \\ \quad - \frac{\alpha}{2 \Delta x} \cdot \Lambda Q_i \cdot \left[\frac{\theta}{(A_i^n)^2} \cdot A_i^n + \frac{Q_{i+1}^n}{A_{i+1}^n} + \frac{Q_i^n}{A_i^n} \right] \\ G' = -\frac{\alpha}{4 \Delta x} (v_i^n + v_{i+1}^n)^2 \cdot \Lambda A_i - \frac{g}{2 \Delta x} \cdot \Sigma A_i \cdot \Lambda z_i - \frac{g}{2} \cdot \Sigma A_i \cdot S F 2 \end{array} \right.$$

3.1.4 Casiers

Un casier est un point de calcul du lit majeur auquel est affecté une capacité de stockage (un volume). Il est décrit par son aire en fonction du niveau d'eau.

Le casier est relié à d'autres casiers ou au lit mineur par un ensemble de tronçons.

Une équation de continuité relie les variations du niveau d'eau y au point casier et les débits échangés avec les points de calcul k qui lui sont connectés. Cette relation suppose formellement un niveau uniforme sur toute la surface S du "casier" :

$$S \cdot \frac{\partial y}{\partial t} = \sum_k Q_k \quad (3.19)$$

3.1.5 Conditions aux limites

A chaque point de calcul qui se trouve à une extrémité du modèle, il faut imposer une condition aux limites.

Carima propose trois types de conditions aux limites :

3.1. UN MODÈLE D'HYDRAULIQUE FLUVIALE MONODIMENSIONNEL : CARIMA

- un débit,
- une hauteur d'eau,
- une courbe de tarage (i.e. une relation entre hauteur et débit).

Cette condition aux limites peut s'exprimer sous la forme :

$$\alpha \Delta z_i + \beta \Delta Q_i = \gamma \quad (3.20)$$

où α , β et γ sont des coefficients à définir en fonction du type de la condition aux limites.

3.1.6 Résolution du système des équations discrétisées

La discrétisation des équations fluviales de Saint Venant dans le cas d'un tronçon entre les points i et $i + 1$ produit alors deux équations linéaires :

$$\begin{array}{ccccccc} A & \Delta z_{i+1} & +B & \Delta Q_{i+1} & = & C & \Delta z_i & +D & \Delta Q_i & + & G \\ A' & \Delta z_{i+1} & +B' & \Delta Q_{i+1} & = & C' & \Delta z_i & +D' & \Delta Q_i & + & G' \end{array}$$

L'algorithme de résolution (dit de double balayage) procède par éliminations successives entre les extrémités du tronçon.

On trouvera une description détaillée de cet algorithme dans l'ouvrage de Cunge [2].

3.1.7 Étude de la stabilité

Afin d'étudier la stabilité du modèle employé afin de résoudre les équations de Saint Venant nous allons utiliser la méthode de Fourier :

Théorème 1 *soit z_j^n la valeur de la discrétisation de la hauteur d'eau au temps n et au point de calcul j et Q_j^n la valeur de la discrétisation du débit au temps n et au point de calcul j .*

Supposons qu'à l'état initial on a

$$\begin{cases} z_j^0 &= A_0 e^{ipj\Delta x} \\ Q_j^0 &= B_0 e^{ipj\Delta x} \end{cases} \quad (3.21)$$

alors il existe une matrice A , indépendante de j et n telle que

$$\begin{cases} z(x, t) &= Z_m e^{ipj\Delta x} \\ Q(x, t) &= Q_m e^{ipj\Delta x} \end{cases} \quad \text{avec} \quad \begin{pmatrix} A_n \\ B_n \end{pmatrix} = A^n \begin{pmatrix} A_0 \\ B_0 \end{pmatrix} \quad (3.22)$$

Remarque : La matrice A est appelée matrice d'amplification. A est une matrice normale et vérifie donc la proposition suivante

$$\|A\|^2 = \rho(A) \quad (3.23)$$

où $\rho(A)$ est le rayon spectral de la matrice A .

Une condition nécessaire de stabilité est donc $\rho(A) \leq 1$. Cette condition est appelée condition de Neumann.

Le schéma de Preissmann est stable tant que $\theta \geq \frac{1}{2}$.

Pour plus de détail dans le développement des calculs on se référera aux articles de Cunge [3], Liggett [13] et Vreugdenhil [20].

3.1.8 Carima en quelques chiffres

Carima est un code de calcul écrit dans les années 70 en Fortran IV. Il compte environ 100 sousroutines et quelques 10000 lignes de code (sans les commentaires).

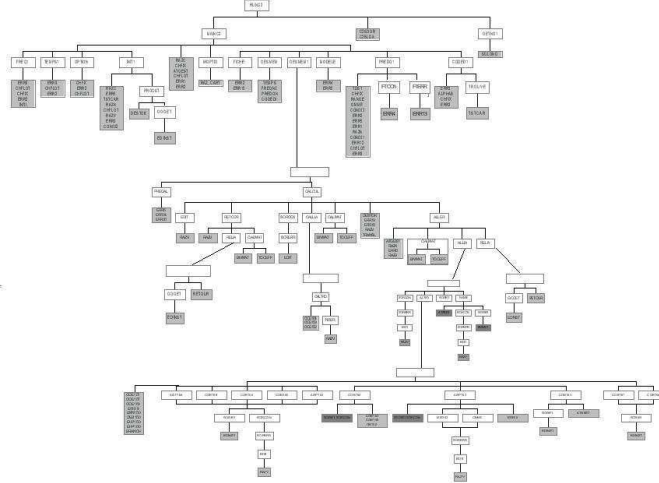


FIG. 3.5 – Organigramme du code de calcul Carima.

Connaître la taille du code ainsi que sa structure permet d'évaluer le travail à effectuer pour dériver le code de calcul.

3.1.9 Construction du modèle Loire

Nous allons présenter dans cette section, la construction d'un modèle hydraulique avec le code Carima.

Nous allons modéliser une partie de la Loire. Situé dans le département du Loiret (45), le val d'Ouzouer (voir la figure 3.6) est localisé à quelques kilomètres en amont d'Orléans.

Depuis 1994, le Plan Loire Grandeur Nature (PLGN) travaille sur le plan d'aménagement de la Loire visant à concilier la sécurité des personnes, la protection de l'environnement et le développement économique. Le Plan Loire modélise entièrement la rivière et son réseau hydraulique grâce au code de calcul Hydra, développé par Hydratec. Ce code est similaire à Carima : il calcule l'évolution de la hauteur d'eau et du débit d'une rivière grâce à la résolution des équations de Saint Venant monodimensionnelles et à partir d'un découpage en casier de la zone inondable il modélise la propagation d'une inondation.

Grâce aux données fournies par le Plan Loire Grandeur Nature, nous allons construire le modèle du Val d'Ouzouer avec Carima.

Coupes transversales définissant le lit de la rivière

Dans un premier temps, nous allons définir le lit de la rivière et de ses affluents grâce à un jeu de coupes transversales.

On décrit ces sections géométriquement avec des couples (Hauteur, Largeur) (cf figure 3.7).

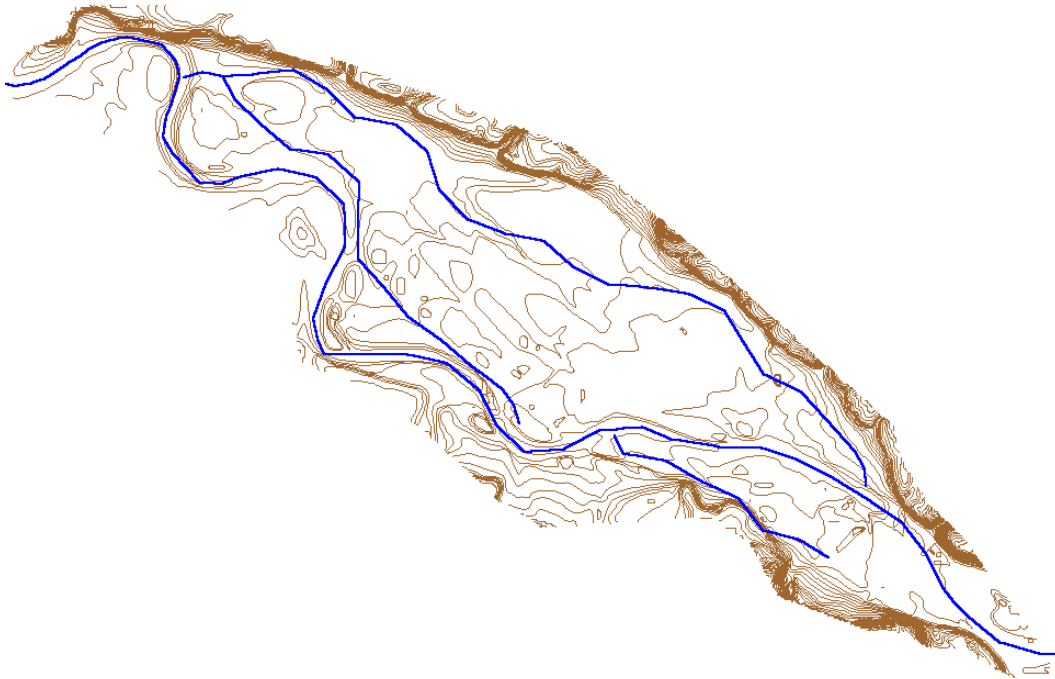


FIG. 3.6 – Topographie mise en relief par les lignes de niveaux décrivant le val d'Ouzouer.

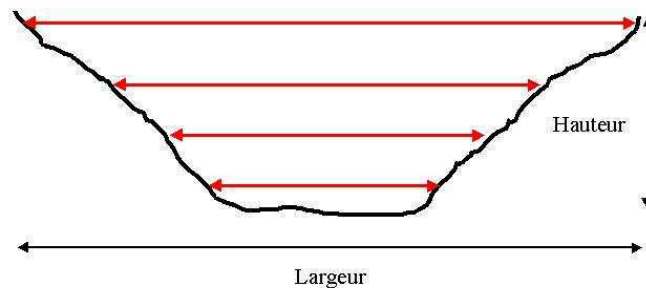


FIG. 3.7 – Description d'une coupe transversale avec le logiciel Carima.

L'ensemble de ces coupes transversales décrit le lit mineur de la rivière et de ses affluents. Dans le cas de notre étude, le réseau hydraulique se découpe ainsi :

- la Loire : 50 sections définissent les 50 km de Loire dans le Val d'Ouzouer,
- le Rousson : 15 sections pour 6 km de cet affluent,
- le déversoir de Saint Martin : 3 sections pour 1 km,
- le déversoir d'Ouzouer : 23 sections pour 20 km,
- la vieille Bonnée : 21 sections pour 13 km.

Une fois ces sections définies, il suffit de définir les liens (ou sections hydrauliques) entre ces points, afin d'indiquer au code de calcul l'enchaînement à suivre entre ces coupes transversales (cf figure 3.8).

Maintenant que la rivière et ses affluents ont été décrits, nous allons voir comment définir la plaine inondable.

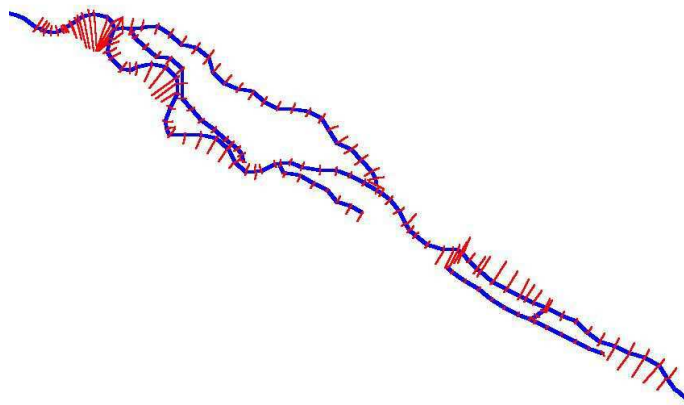


FIG. 3.8 – *Découpage de La Loire en coupe transversales et sections hydrauliques avec le code de calcul Carima.*

Casier définissant la plaine inondable

La modélisation par casiers (ou cellules) dans le champ d'inondation est une procédure intermédiaire entre la modélisation unidimensionnelle pure et une modélisation bidimensionnelle.

Le découpage d'une plaine inondable en casiers nécessite une étude hydraulique minutieuse. La description du découpage du Val d'Ouzouer a été fournie par le Plan Loire Grandeur Nature (voir la figure 3.9).

Un casier est un volume de stockage défini par des couples (hauteur, surface). Dans le cadre de cette étude, le Val se découpe comme suit :

- le Val de Gien : 10 casiers,
- le Val de Dampierre : 2 casiers,
- le Val de Sully : 5 casiers,
- le Val d'Ouzouer : 21 casiers,
- Châteauneuf sur Loire : 2 casiers.

Il suffit maintenant de définir les liens entre les casiers afin d'assurer l'échange d'eau entre eux.

On relie le lit de la rivière à la plaine inondable

Il ne nous reste plus qu'à relier la rivière avec la plaine inondable. Cette étape nécessite une étude minutieuse de la rivière afin de déterminer les zones où le cours d'eau est susceptible de sortir de son lit. En effet, c'est l'utilisateur qui va déterminer les endroits où la rivière va déborder, c'est-à-dire qu'il va falloir positionner les liens entre la rivière et les casiers (voir figure 3.10).

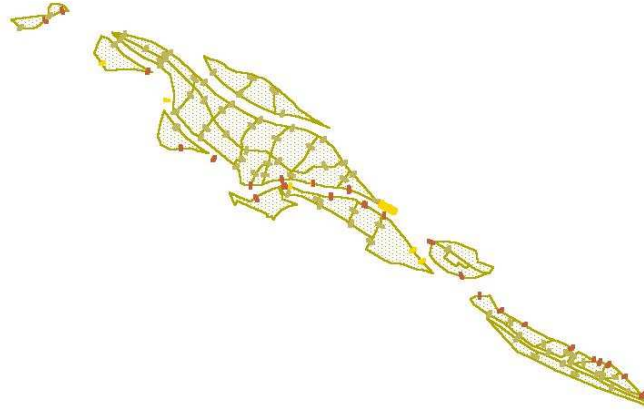


FIG. 3.9 – Découpage du Val d'Ouzouer en casiers avec le code de calcul Carima.

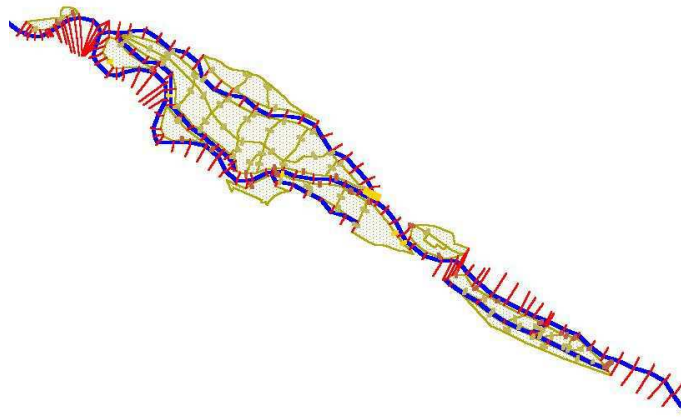


FIG. 3.10 – Modélisation du Val d'Ouzouer avec le code de calcul Carima.

Les conditions aux limites

Les conditions aux limites vont nous permettent d'étudier différents scénarii de crues. En accord avec le Plan Loire Grandeur Nature, nous avons décidé de travailler sur 6 scénarii de crues (cf figure 3.11) :

- période de retour de 50 ans qui correspond à un pic de crue de $4200m^3/s$,
- période de retour de 70 ans qui correspond à un pic de crue de $5000m^3/s$,
- période de retour de 100 ans qui correspond à un pic de crue de $6000m^3/s$,
- période de retour de 170 ans qui correspond à un pic de crue de $6500m^3/s$,
- période de retour de 200 ans qui correspond à un pic de crue de $7000m^3/s$,
- période de retour de 500 ans qui correspond à un pic de crue de $8500m^3/s$.

La condition initiale

Il nous reste à définir une condition initiale correspondant à chacun des scénarii présentés ci-dessus. Pour cela, nous devons prendre une ligne d'eau uniforme sur toute la rivière et, progressivement, baisser le niveau jusqu'à ce que l'on obtienne les valeurs désirées aux limites. Il suffit alors de récupérer les valeurs obtenues en chaque point de la rivière et de les utiliser pour initialiser l'exécution du code.

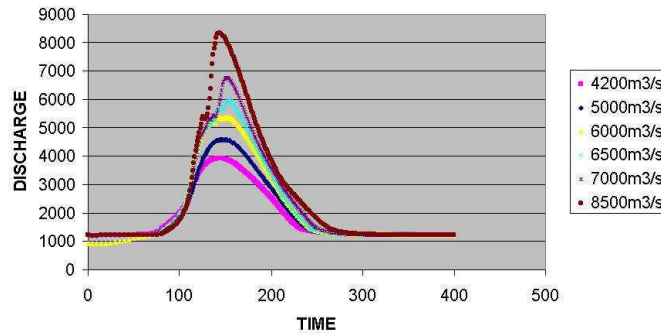


FIG. 3.11 – Scénarii utilisés dans l'étude du Val d'Ouzouer avec Carima.

Remarque : La pente du fond du lit de la rivière ne doit pas être trop importante afin d'éviter que l'écoulement soit torrentiel, ce qui serait en contradiction avec les hypothèses permettant d'écrire les équations de Saint Venant (i.e. les accélérations verticales doivent être négligeable). Afin d'éviter cette accélération de l'écoulement, nous avons la possibilité de remplacer cette section de la rivière trop pentue par un déversoir. L'écoulement au dessus de ce déversoir n'est plus régie par les équations de Saint Venant mais par une relation entre le débit et la hauteur.

Résultats

Une fois le modèle construit, nous avons testé plusieurs scénarii afin d'étudier le comportement de l'écoulement dans le Val d'Ouzouer.

Pour un scénario de période de retour 200 ans, par exemple, nous allons simuler une crue d'une durée de 400 heures (le pas de temps est de 30 minutes).

Nous pouvons voir sur la figure 3.12 qu'une grande partie du Val d'Ouzouer sera inondée après 200 heures, c'est-à-dire au moment du pic de crue.

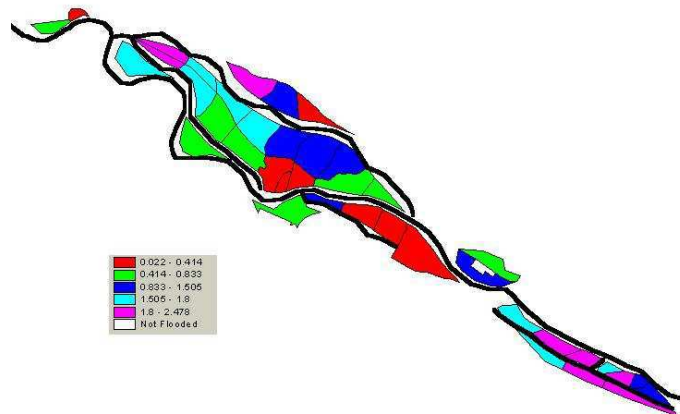


FIG. 3.12 – Inondation du Val d'Ouzouer pour un scénario de période de retour 200 ans modélisé avec le code de calcul Carima.

L'inondation du Val d'Ouzouer permettra de casser le pic de crue afin de protéger la ville d'Orléans.

3.1.10 Conclusion

Dans cette section, nous avons étudié le modèle d'hydraulique fluviale, Carima. Basé sur la résolution des équations de Saint Venant monodimensionnelles, il permet de prendre en compte le débordement dans la plaine inondable grâce à un découpage de celle-ci en casiers.

Ensuite, un exemple de modélisation avec Carima a été présenté. Cette modélisation, bien que construite par Hydratec, nous permet d'appréhender les difficultés inhérentes à la modélisation 1D :

- *Déterminer la localisation des coupes transversales afin de définir au mieux la rivière.* L'ensemble des coupes transversales doit prendre en compte la totalité des particularités de la rivière pouvant influencer le comportement de l'écoulement (virage, rétrécissement, élargissement, ressaut, etc ...).
- *Définir la géométrie de la section.* La définition géométrique de la section devra alors être mesurée par un géomètre. Mais cette option peut s'avérer être très chère. Nous verrons ultérieurement, un exemple d'utilisation de l'assimilation de données afin d'estimer le relief équivalent (cet exemple sera mené avec LMCFLD, un logiciel de modélisation bidimensionnelle, mais pourrait être appliqué de la même manière avec Carima).
- *Déterminer le coefficient de rugosité de la section.* Il faut définir un coefficient de Strickler capable de représenter globalement le frottement sur la section. Généralement, ce coefficient est défini manuellement grâce à une succession d'exécution du code direct. L'assimilation de donnée fournit un outil pour faire cette estimation de manière automatique.
- *Créer le jeu de casiers.* Le découpage de la plaine inondable en casier requiert une expertise minutieuse de la zone. Il n'existe pas vraiment de règle à respecter pour découper le domaine. Il faut déterminer l'ensemble des points bas du domaine, puis définir le sens global de l'écoulement. A partir de ces deux données, le découpage va évoluer en fonction du but de l'étude hydraulique (écoulement général, impact financier, etc ...).
- *Déterminer les zones de débordements.* La modélisation 1D nécessite de connaître *a priori* les zones de la berge susceptible d'être noyées. Cette étude doit aussi prendre en compte les différents scénarii à tester comme par exemple l'ouverture d'un brèche dans une digue.
- *Déterminer la condition initiale.* Cette donnée pose le problème de la stabilité du modèle. Afin de déterminer une condition initiale, il faut tout d'abord imposer une ligne d'eau identique en chaque point de calcul, puis baisser progressivement ce niveau jusqu'à obtenir les valeurs souhaités. L'estimation de la condition initiale est un enjeu primordial pour la prévision. L'assimilation de donnée est un outil qui permet de calculer une condition initiale optimale en fonction d'un jeu d'observations données. Nous verrons dans la suite quelques exemples d'estimation de conditions initiales.
- *Déterminer une condition aux limites.* Déterminer une condition aux limites c'est définir le scénario d'une crue. A partir de statistiques de crue passée, il faut reconstituer la condition aux limites responsable de cet événement. Puis, suivant les nouveaux scénario que l'on veut tester, il faut reconstituer la condition aux limites répondant

aux attentes de l'expérimentateur. Encore ici, l'assimilation de données fournit un outil performant pour déterminer cette condition aux limites comme nous le verrons plus tard.

3.2 Une maquette de calcul des écoulements bidimensionnel : LMCFLD

Le modèle LMCFLD a été développé au Laboratoire de Modélisation et Calcul (LMC) de Grenoble par Yang [24]. La vocation de ce modèle est de tester diverses méthodes mathématiques sur des exemples schématiques.

3.2.1 Les équations de Saint Venant

Les équations de base que nous utiliserons dans LMCFLD sont les équations de Saint Venant bidimensionnelles qui décrivent la conservation de la masse et la quantité de mouvement dans le fluide.

Dans leur formulation hauteur-vitesse, elles se résument ainsi :

- équation de conservation de la masse :

$$\frac{\partial Z}{\partial t} + \frac{\partial(hu)}{\partial x} + \frac{\partial(hv)}{\partial y} = 0 \quad (3.24)$$

- équation de la quantité de mouvement :

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + g \frac{\partial Z}{\partial x} - f v = -g \frac{u \sqrt{u^2 + v^2}}{C^2 h} + \gamma_t \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad (3.25)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + g \frac{\partial Z}{\partial y} + f u = -g \frac{v \sqrt{u^2 + v^2}}{C^2 h} + \gamma_t \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \quad (3.26)$$

avec

- h : la profondeur de la couche liquide, avec la contrainte qu'elle est toujours positive,
- u et v : les composantes de la vitesse, respectivement suivant les directions x (ouest-est) et y (sud-nord),
- Z : la cote de la surface libre, ainsi $h = Z - Z_b$ lorsque Z_b est la cote du lit de la rivière,
- C : le coefficient de Chezy calculé par l'équation suivante,

$$C = \frac{1}{n} h^{1/6} \quad (3.27)$$

où n est le coefficient de Manning exprimant la rugosité,

- γ_t : le coefficient de viscosité turbulente,
- f : le paramètre de Coriolis,
- g : la pesanteur.

Remarque : Le paramètre de Coriolis influence le comportement de l'écoulement dans un domaine de grande taille. Bien que les exemples que nous traiterons ultérieurement ne modélisent pas de si grands domaines, dans un souci de généralisation, nous présentons ici les équations complètes de Saint Venant.

3.2.2 Les conditions aux limites

On considère le domaine Ω dont la frontière est composée de trois parties $\partial\Omega_1$, $\partial\Omega_2$, $\partial\Omega_3$, qui représentent respectivement l'écoulement entrant à l'ouest, les parois latérales nord et sud et la sortie du fluide par la paroi est.

On suppose que la rivière s'écoule d'ouest en est avec des bords latéraux au nord et au sud.

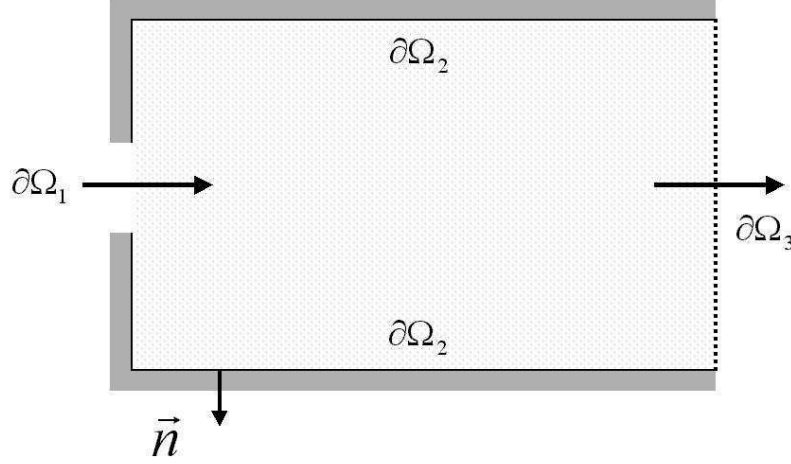


FIG. 3.13 – Composition du domaine Ω considéré par LMCFLD.

Dans la paroi ouest se trouve une brèche par laquelle l'eau va s'écouler. On donne une hauteur d'eau dans cette brèche .

$$h = h_1$$

Les conditions aux limites nord et sud sont de type glissement sans frottement :

$$\vec{u} \cdot \vec{n} = 0$$

et la condition aux limites est est de type frontière ouverte :

$$\frac{\partial \vec{u}}{\partial \vec{n}} = 0$$

Au temps initial, on donne sur tout le domaine une vitesse de l'écoulement \vec{u} ainsi qu'une profondeur h .

$$\vec{u}(t = 0) = \vec{u}_0 \quad h(t = 0) = h_0$$

On est ramené à un système d'équations non linéaires avec conditions aux limites et condition initiale. Il reste à prouver mathématiquement que le problème est bien posé (existence, unicité et stabilité), cette démonstration peut se trouver dans la thèse de Yang [24].

3.2.3 Changement des variables pour les équations de Saint-Venant

Avant de discrétiser les équations de Saint Venant, nous allons introduire un changement de variables. Cette approche s'inspire des travaux de Zeng[25]. Ce changement de

variable permet d'assurer la conservation de la masse et de l'énergie du système discrétisé que nous introduirons par la suite (cf la thèse de Yang [24] pour plus de renseignement sur les développements théoriques).

Posons

$$\begin{cases} \Phi &= \sqrt{gh} \\ U &= u\Phi \\ V &= v\Phi \\ \phi &= gZ \end{cases}$$

Les équations (3.24) et (3.25) s'expriment avec ces variables de la manière suivante :

$$\begin{aligned} \frac{\partial U}{\partial t} + \left(\frac{\partial(uU)}{\partial x} - \frac{U}{2} \frac{\partial u}{\partial x} \right) + \left(\frac{\partial(vU)}{\partial y} - \frac{U}{2} \frac{\partial v}{\partial y} \right) &= \\ fV - \Phi \frac{\partial \phi}{\partial x} + \Phi \cdot \gamma_t \cdot \Delta \frac{U}{\Phi} - kU \\ \frac{\partial V}{\partial t} + \left(\frac{\partial(uV)}{\partial x} - \frac{V}{2} \frac{\partial u}{\partial x} \right) + \left(\frac{\partial(vV)}{\partial y} - \frac{V}{2} \frac{\partial v}{\partial y} \right) &= \\ -fU - \Phi \frac{\partial \phi}{\partial y} + \Phi \cdot \gamma_t \cdot \Delta \frac{V}{\Phi} - kV \\ \frac{\partial \phi}{\partial t} + \frac{\partial(U\Phi)}{\partial x} + \frac{\partial(V\Phi)}{\partial y} &= 0 \end{aligned}$$

avec $k = g^2 C_D \frac{\sqrt{U^2 + V^2}}{\Phi^3}$.

Suite à ce changement de variable, nous allons appliquer un schéma aux différences finies pour discrétiser les équations (3.28), ce schéma assurera les propriétés suivantes :

- Les équations discrétisées possèdent la même propriété de conservation que les équations différentielles continues.
- Le schéma de discrétisation à l'avantage d'être conservatif en énergie.

3.2.4 Discrétisation des équations de Saint Venant

Choix du maillage

Avant de discrétiser en espace les équations considérées avec une méthode de différences finies, on définit d'abord les variables physiques sur un maillage rectangulaire de type ARAKAWA, défini comme suit :

- Dans le plan $(0xy)$, les coordonnées des points de grille sont définie par :

$$\begin{aligned} x_I &= I\Delta x \quad \forall I = 0, \dots, Max \\ y_J &= J\Delta y \quad \forall J = 0, \dots, NMax \end{aligned}$$

Par conséquent, la valeur de toute variable s'écrit :

$$X(x_I, y_J) = X_J^I$$

- Dans ce maillage, on définit trois types de point :

h-point défini par les coordonnées entières (x_I, y_J) , $\forall I = 0, \dots, Max$ et $\forall J = 0, \dots, NMax$.

Ce sont les points où l'on calcule les approximations des variables h , ϕ , Φ et Z_b .

3.2. UNE MAQUETTE DE CALCUL DES ÉCOULEMENTS BIDIMENSIONNEL : LMCFLD

u-point défini par les coordonnées $(x_{I+1/2}, y_J), \forall I = 0, \dots, Max$ et $\forall J = 0, \dots, NMax$.
Ce sont les points où l'on calcule les approximations des variables u et U .

v-point défini par les coordonnées $(x_I, y_{J+1/2}), \forall I = 0, \dots, Max$ et $\forall J = 0, \dots, NMax$.
Ce sont les points où l'on calcule les approximations des variables v et V .

Discrétisation spatiale

Sur la grille précédente nous allons discrétiser le système hydraulique en espace par un schéma de différences finies. Nous utiliserons le même pas d'espace dans les deux directions x et y . Puisque les variables sont situées de manière alternée sur le maillage (voir figure 3.14) et pour éviter les abus de langage, on exprime la discrétisation des dérivées partielles par rapport aux coordonnées spatiales sans distinguer les indices entiers et demi-entiers. Nous posons δh la discrétisation en espace.

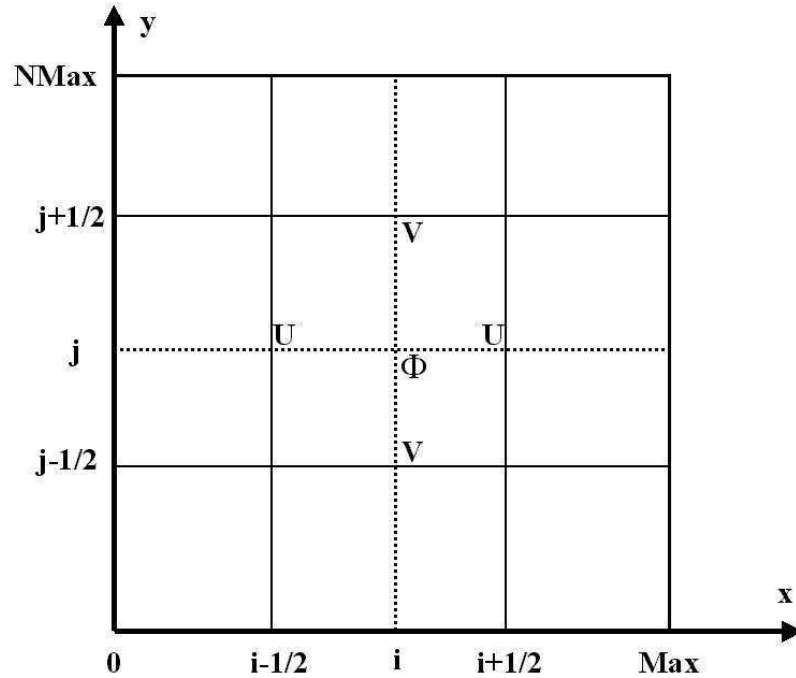


FIG. 3.14 – Distribution des variables dans un maillage de type ARAKAWA. Cette distribution est utilisée pour discrétiser les équations de Saint Venant dans le logiciel LMCFLD.

Si l'on pose

$$\begin{cases} I = i + \frac{1}{2} \text{ ou } i & \forall i = 0, \dots, Max \\ J = j + \frac{1}{2} \text{ ou } j & \forall j = 0, \dots, NMax \end{cases}$$

La discrétisation spatiale utilisée sur les équations de Saint Venant est basée sur les règles suivantes :

- les dérivées par rapport à x de toutes fonction f se discrétisent comme suit :

$$\left(\frac{\partial f}{\partial x} \right)_{I,J} \approx \frac{f_J^{I+1/2} - f_J^{I-1/2}}{\delta h} \quad (3.28)$$

- les dérivées par rapport à y de toutes fonction f se discrétisent comme suit :

$$\left(\frac{\partial f}{\partial y}\right)_{I,J} \approx \frac{f_{J+1/2}^I - f_{J-1/2}^I}{\delta h} \quad (3.29)$$

- l'opérateur Laplacien est approché par le schéma classique à cinq points suivant :

$$(\underline{\Delta}X)_{I,J} \approx \frac{X_J^{I+1} + X_J^{I-1} + X_{J+1}^I + X_{J-1}^I - 4X_J^I}{\delta h^2} \quad (3.30)$$

On rappelle d'abord les équations de notre système (3.28) :

$$\begin{aligned} \frac{\partial U}{\partial t} + \left(\frac{\partial(uU)}{\partial x} - \frac{U}{2} \frac{\partial u}{\partial x}\right) + \left(\frac{\partial(vU)}{\partial y} - \frac{U}{2} \frac{\partial v}{\partial y}\right) &= \\ fV - \Phi \frac{\partial \phi}{\partial x} + \Phi \cdot \gamma_t \cdot \Delta \frac{U}{\Phi} - kU \\ \frac{\partial V}{\partial t} + \left(\frac{\partial(uV)}{\partial x} - \frac{V}{2} \frac{\partial u}{\partial x}\right) + \left(\frac{\partial(vV)}{\partial y} - \frac{V}{2} \frac{\partial v}{\partial y}\right) &= \\ -fU - \Phi \frac{\partial \phi}{\partial y} + \Phi \cdot \gamma_t \cdot \Delta \frac{V}{\Phi} - kV \\ \frac{\partial \phi}{\partial t} + \frac{\partial(U\Phi)}{\partial x} + \frac{\partial(V\Phi)}{\partial y} &= 0 \end{aligned}$$

avec $k = g^2 C_D \frac{\sqrt{U^2 + V^2}}{\Phi^3}$

Mais avant d'appliquer les règles (3.28), (3.29) et (3.30) au système précédent, nous allons moyenner certaines variables en respectant les règles suivantes :

- certaines variables seront moyennées suivant x :

$$f^x \rightarrow \frac{f_J^{I+1/2} + f_J^{I-1/2}}{2} \quad (3.31)$$

- certaines variables seront moyennées suivant y :

$$f^y \rightarrow \frac{f_{J+1/2}^I + f_{J-1/2}^I}{2} \quad (3.32)$$

- certaines variables seront moyennées suivant x et y :

$$f^{xy} \rightarrow \frac{f_{J+1/2}^{I+1/2} + f_{J+1/2}^{I-1/2} + f_{J-1/2}^{I+1/2} + f_{J-1/2}^{I-1/2}}{4} \quad (3.33)$$

Dans le système suivant, nous mettons en relief les variables qui seront moyennées :

$$\begin{aligned} \frac{\partial \phi}{\partial t} + \frac{\partial(U\Phi^x)}{\partial x} + \frac{\partial(V\Phi^y)}{\partial y} &= 0 \\ \frac{\partial U}{\partial t} + \left(\frac{\partial(u^x U^x)}{\partial x} - \frac{U}{2} \frac{\partial u^x}{\partial x}\right) + \left(\frac{\partial(v^x U^y)}{\partial y} - \frac{U}{2} \frac{\partial v^x}{\partial y}\right) &= \end{aligned}$$

$$\begin{aligned}
 & fV^{xy} - \Phi^x \frac{\partial \phi}{\partial x} + \Phi^x \cdot \gamma_t \cdot \Delta \frac{U}{\Phi^x} - k_x U \\
 & \frac{\partial V}{\partial t} + \left(\frac{\partial(u^y V^x)}{\partial x} - \frac{V}{2} \frac{\partial u^y}{\partial x} \right) + \left(\frac{\partial(v^y V^y)}{\partial y} - \frac{V}{2} \frac{\partial v^y}{\partial y} \right) = \\
 & -fU^{xy} - \Phi^y \frac{\partial \phi}{\partial y} + \Phi^y \cdot \gamma_t \cdot \Delta \frac{V}{\Phi^y} - k_y V
 \end{aligned} \tag{3.34}$$

avec $k_x = g^2 C_D \frac{\sqrt{U^2 + (V^{xy})^2}}{(\Phi^x)^3}$ et $k_y = g^2 C_D \frac{\sqrt{(U^{xy})^2 + V^2}}{(\Phi^y)^3}$.

Grâce aux indications présentées précédemment, on discrétise les équations de Saint Venant en espace à l'intérieur du domaine.

- discrétisation de la première équation :

$$\begin{aligned}
 & \frac{\partial \phi}{\partial t} + \frac{\partial(U\Phi^x)}{\partial x} + \frac{\partial(V\Phi^y)}{\partial y} = 0 \\
 & \Leftrightarrow \\
 & \left[\frac{\partial \phi}{\partial t} \right]_J^I + \frac{(U\Phi^x)_J^{I+1/2} - (U\Phi^x)_J^{I-1/2}}{\delta h} + \frac{(V\Phi^y)_{J+1/2}^I - (V\Phi^y)_{J-1/2}^I}{\delta h} = 0
 \end{aligned}$$

- discrétisation de la deuxième équation :

$$\begin{aligned}
 & \frac{\partial U}{\partial t} + \left(\frac{\partial(u^x U^x)}{\partial x} - \frac{U}{2} \frac{\partial u^x}{\partial x} \right) + \left(\frac{\partial(v^x U^y)}{\partial y} - \frac{U}{2} \frac{\partial v^x}{\partial y} \right) = \\
 & fV^{xy} - \Phi^x \frac{\partial \phi}{\partial x} + \Phi^x \cdot \gamma_t \cdot \Delta \frac{U}{\Phi^x} - k_x U \\
 & \Leftrightarrow \\
 & \left[\frac{\partial U}{\partial t} \right]_J^I + \left(\frac{(u^x U^x)_J^{I+1/2} - (u^x U^x)_J^{I-1/2}}{\delta h} - \left[\frac{U}{2} \right]_J^I \frac{(u^x)_J^{I+1/2} - (u^x)_J^{I-1/2}}{\delta h} \right) \\
 & + \left(\frac{(v^x U^y)_{J+1/2}^I - (v^x U^y)_{J-1/2}^I}{\delta h} - \left[\frac{U}{2} \right]_J^I \frac{(v^x)_{J+1/2}^I - (v^x)_{J-1/2}^I}{\delta h} \right) = \\
 & \frac{[fV^{xy}]_J^I - [\Phi^x]_J^I \frac{\phi_J^{I+1/2} - \phi_J^{I-1/2}}{\delta h} - [k_x U]_J^I}{\delta h^2} \\
 & + [\Phi^x]_J^I \cdot \frac{\gamma_t}{\delta h^2} \cdot \left[\left(\frac{U}{\Phi^x} \right)_J^{I+1} + \left(\frac{U}{\Phi^x} \right)_J^{I-1} + \left(\frac{U}{\Phi^x} \right)_{J+1}^I + \left(\frac{U}{\Phi^x} \right)_{J-1}^I - 4 \left(\frac{U}{\Phi^x} \right)_J^I \right] \\
 & \text{avec } k_x = g^2 C_D \frac{2\sqrt{U^2 + (V^{xy})^2}}{(\Phi^x)^3}
 \end{aligned}$$

- Discrétisation de la troisième équation :

$$\begin{aligned}
 & \frac{\partial V}{\partial t} + \left(\frac{\partial(u^y V^x)}{\partial x} - \frac{V}{2} \frac{\partial u^y}{\partial x} \right) + \left(\frac{\partial(v^y V^y)}{\partial y} - \frac{V}{2} \frac{\partial v^y}{\partial y} \right) = \\
 & -fU^{xy} - \Phi^y \frac{\partial \phi}{\partial y} + \Phi^y \cdot \gamma_t \cdot \Delta \frac{V}{\Phi^y} - k_y V \\
 & \Leftrightarrow \\
 & \left[\frac{\partial V}{\partial t} \right]_J^I + \left(\frac{(u^y V^x)_J^{I+1/2} - (u^y V^x)_J^{I-1/2}}{\delta h} - \left[\frac{V}{2} \right]_J^I \frac{(u^y)_J^{I+1/2} - (u^y)_J^{I-1/2}}{\delta h} \right) \\
 & + \left(\frac{(v^y V^y)_{J+1/2}^I - (v^y V^y)_{J-1/2}^I}{\delta h} - \left[\frac{V}{2} \right]_J^I \frac{(v^y)_{J+1/2}^I - (v^y)_{J-1/2}^I}{\delta h} \right) = \\
 & \frac{-[fU^{xy}]_J^I - [\Phi^y]_J^I \frac{\phi_{J+1/2}^I - \phi_{J-1/2}^I}{\delta h} - [k_y V]_J^I}{\delta h^2} \\
 & + [\Phi^y]_J^I \cdot \frac{\gamma_t}{\delta h^2} \cdot \left[\left(\frac{V}{\Phi^y} \right)_J^{I+1} + \left(\frac{V}{\Phi^y} \right)_J^{I-1} + \left(\frac{V}{\Phi^y} \right)_{J+1}^I + \left(\frac{V}{\Phi^y} \right)_{J-1}^I - 4 \left(\frac{V}{\Phi^y} \right)_J^I \right]
 \end{aligned}$$

$$\text{avec } k_y = g^2 C_D \frac{\sqrt{(U^{xy})^2 + V^2}}{(\Phi^y)^3}$$

Discrétisation temporelle

Soit les séquences de temps suivantes :

$$t_n = n\Delta t \quad \forall n \geq 0$$

où δt est le pas de temps.

Soit X^n l'approximation de X au temps n , nous utilisons alors le schéma explicite de leap-frog pour la discrétisation en temps :

$$\frac{\partial X}{\partial t} = \frac{X^{n+1} - X^{n-1}}{\Delta t} \quad (3.35)$$

3.2.5 Résolution du système des équations discrétisées

Pour résoudre le système d'équations dynamiques (3.24) et (3.25), on va utiliser la méthode à pas fractionnaire introduite par Yanenko [23] et Marchuk [15]. Il s'agit d'une variante des méthodes dites des directions alternées, pour lesquelles on peut consulter les travaux de Douglas et Rachford [4].

Schémas à pas fractionnaire

Cette méthode s'inspire de la méthode de décomposition des opérateurs.

Considérons le système suivant :

$$\begin{cases} \frac{\partial X}{\partial t} + \mathcal{F}(X) = \mathcal{G} \\ X(t=0) = X^0 \end{cases} \quad (3.36)$$

où \mathcal{F} est un opérateur non-linéaire dépendant de la solution X .

Supposons que l'opérateur différentiel défini précédemment (3.36), admette une décomposition naturelle :

$$\mathcal{F}(X) = \sum_{i=1}^M \mathcal{F}_i(X) \quad (3.37)$$

Cette décomposition va nous emmener à décomposer de même le calcul.

Décomposition des équations de Saint Venant

On décompose le système dynamique en trois étapes :

- le processus géostrophique,
- le processus d'advection,
- le processus de dissipation.

Soit l'intervalle de temps $[t_n, t_{n+1}]$ et $X^n = (U^n, V^n, \phi^n)$.

On détermine la valeur de X^{n+1} en appliquant le schéma suivant :

a) le processus géostrophique

$$\begin{aligned}
\left[\frac{\partial U}{\partial t} \right]_J^{I+1/2} &= [fV^{xy}]_J^{I+1/2} - [\Phi^x]_J^{I+1/2} \frac{\phi_J^{I+1/2} - \phi_J^{I-1/2}}{\delta h} \\
\left[\frac{\partial V}{\partial t} \right]_{J+1/2}^I &= -[fU^{xy}]_J^{I+1/2} - [\Phi^y]_J^{I+1/2} \frac{\phi_{J+1/2}^I - \phi_{J-1/2}^I}{\delta h} \\
\left[\frac{\partial \phi}{\partial t} \right]_J^I &= \frac{(U\Phi^x)_J^{I+1/2} - (U\Phi^x)_J^{I-1/2}}{\delta h} + \frac{(V\Phi^y)_{J+1/2}^I - (V\Phi^y)_{J-1/2}^I}{\delta h}
\end{aligned} \tag{3.38}$$

qui donne $U^I = U(t_{n+1})$, $V^I = V(t_{n+1})$ et $\phi^I = \phi(t_{n+1})$.

b) le processus d'advection

$$\begin{aligned}
\left[\frac{\partial U}{\partial t} \right]_J^{I+1/2} &= \left(\frac{(u^x U^x)_J^{I+1/2} - (u^x U^x)_J^{I-1/2}}{\delta h} - \left[\frac{U}{2} \right]_J^{I+1/2} \frac{(u^x)_J^{I+1/2} - (u^x)_J^{I-1/2}}{\delta h} \right) \\
&+ \left(\frac{(v^x U^y)_{J+1/2}^I - (v^x U^y)_{J-1/2}^I}{\delta h} - \left[\frac{U}{2} \right]_j^{I+1/2} \frac{(v^x)_{J+1/2}^I - (v^x)_{J-1/2}^I}{\delta h} \right) \\
\left[\frac{\partial V}{\partial t} \right]_{J+1/2}^I &= \left(\frac{(u^y V^x)_J^{I+1/2} - (u^y V^x)_J^{I-1/2}}{\delta h} - \left[\frac{V}{2} \right]_J^I \frac{(u^y)_J^{I+1/2} - (u^y)_J^{I-1/2}}{\delta h} \right) \\
&+ \left(\frac{(v^y V^y)_{J+1/2}^I - (v^y V^y)_{J-1/2}^I}{\delta h} - \left[\frac{V}{2} \right]_J^I \frac{(v^y)_{J+1/2}^I - (v^y)_{J-1/2}^I}{\delta h} \right) \\
\left[\frac{\partial \phi}{\partial t} \right]_J^I &= 0
\end{aligned} \tag{3.39}$$

qui donne $U^{II} = U(t_{n+1})$, $V^{II} = V(t_{n+1})$ et $\phi^{II} = \phi(t_{n+1})$. Les conditions initiales sont celles données par le processus géostrophique.

c) le processus de dissipation

$$\begin{aligned}
\left[\frac{\partial U}{\partial t} \right]_J^{I+1/2} &= [\Phi^x]_J^{I+1/2} \cdot \frac{\gamma_t}{\delta h^2} \cdot \left[\left(\frac{U}{\Phi^x} \right)_J^{I+1} + \left(\frac{U}{\Phi^x} \right)_J^{I-1} \right] \\
&+ [\Phi^x]_J^{I+1/2} \cdot \frac{\gamma_t}{\delta h^2} \cdot \left[\left(\frac{U}{\Phi^x} \right)_{J+1}^I + \left(\frac{U}{\Phi^x} \right)_{J-1}^I - 4 \left(\frac{U}{\Phi^x} \right)_J^I \right] - [k_x U]_J^I \\
\left[\frac{\partial V}{\partial t} \right]_{J+1/2}^I &= [\Phi^y]_J^{I+1/2} \cdot \frac{\gamma_t}{\delta h^2} \cdot \left[\left(\frac{V}{\Phi^y} \right)_J^{I+1} + \left(\frac{V}{\Phi^y} \right)_J^{I-1} \right] \\
&+ [\Phi^y]_J^{I+1/2} \cdot \frac{\gamma_t}{\delta h^2} \cdot \left[\left(\frac{V}{\Phi^y} \right)_{J+1}^I + \left(\frac{V}{\Phi^y} \right)_{J-1}^I - 4 \left(\frac{V}{\Phi^y} \right)_J^I \right] - [k_y V]_J^I \\
\left[\frac{\partial \phi}{\partial t} \right]_J^I &= 0
\end{aligned} \tag{3.40}$$

qui donne $U^{n+1} = U(t_{n+1})$, $V^{n+1} = V(t_{n+1})$ et $\phi^{n+1} = \phi(t_{n+1})$.

Remarque : Les valeurs de u , v et Φ doivent être modifiées à la fin de chaque processus par le changement de variable :

$$\left\{ \begin{array}{lcl} u_j^{i+1/2} & = & \left[\frac{U}{\Phi^x} \right]_j^{i+1/2} = \frac{2U_j^{i+1/2}}{\Phi_j^{i+1} + \Phi_j^i} \\ v_{j+1/2}^i & = & \left[\frac{V}{\Phi^y} \right]_{j+1/2}^i = \frac{2V_{j+1/2}^i}{\Phi_{j+1}^i + \Phi_j^i} \\ \Phi_j^i & = & \left[\sqrt{\phi - gZ_b} \right]_j^i \end{array} \right.$$

Sachant qu'il s'agit de trois processus de vitesses différentes, on choisit des pas de temps différents pour chaque processus. En général, le processus géostrophique varie plus rapidement que les deux autres. Pour permettre un calcul plus précis, on choisit un pas de temps pour les processus d'advection et de dissipation Δt_1 et on choisit un pas de temps Δt_2 plus petit pour le processus géostrophique :

$$\Delta t_2 = \frac{\Delta t_1}{N_g}$$

où $N_g \in \mathbf{N}$.

En résumé, pour résoudre le système d'équations de Saint Venant, nous découpons le processus en trois étapes. Cette décomposition nous permet de travailler avec un schéma explicite, gagnant ainsi en temps de calcul.

3.2.6 Mise en place du modèle

LMCFLD calcule l'évolution de l'écoulement dans un bassin rectangulaire. Le seul apport d'eau dans ce bassin provient de la façade ouest dans laquelle se trouve une brèche (à $X = 0$).

Afin de décrire ce bassin, on va tout d'abord définir la grille du domaine en fixant le nombre de points de calculs en X et en Y .

Ensuite dans un fichier *Relief.dat*, on définit pour chaque nœud de la grille le niveau du relief.

3.2. UNE MAQUETTE DE CALCUL DES ÉCOULEMENTS BIDIMENSIONNEL : LMCFLD

(X)	(Y)	Niveau (m)
	...	
12	46	10.00
12	47	10.00
12	48	10.00
12	49	10.00
12	50	10.00
12	51	10.00
12	52	10.00
12	53	10.00
12	54	10.00
13	0	10.00
13	1	10.00
13	2	10.00
13	3	10.00
13	4	10.00
	...	

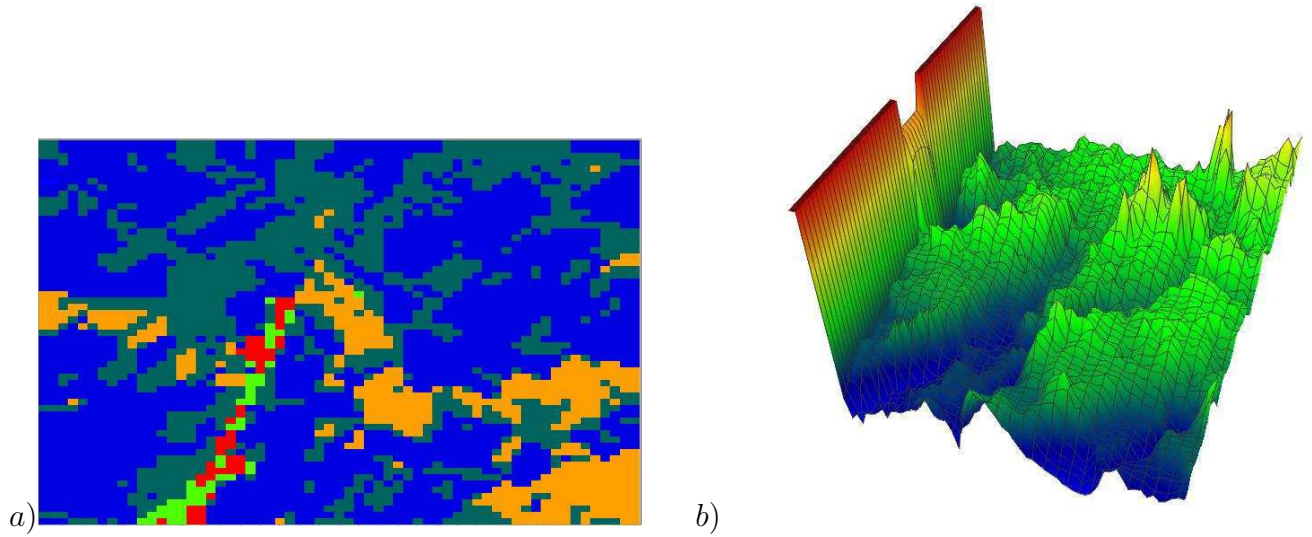
Puis on va définir les coefficients de rugosité (coefficient de Manning) pour chaque nœud de la grille dans un fichier *Manning.dat*.

(X)	(Y)	Manning
	...	
15	55	7
15	56	2
15	57	2
15	58	2
15	59	2
15	60	7
16	0	2
16	1	2
	...	

Puis on définit la condition aux limites imposée dans la brèche, grâce au fichier *LimitCondition.CL* dans lequel pour chaque pas de temps on donne la hauteur d'eau imposée.

Hauteur (m)	Temps (s)
	...
5.	44.
5.	46.
5.	48.
5.	50.
5.	52.
5.	54.
5.	56.
5.	58.
	...

Enfin, dans le fichier *data.txt* sont définis successivement le pas d'espace en x , celui en y , le pas de temps, le temps de calcul, la hauteur d'eau initiale dans la brèche et la hauteur



TAB. 3.1 – a) Exemple de représentation des coefficients de Manning dans LMCFLD . b) Exemple de relief pris en compte dans LMCFLD.

d'eau initiale dans le bassin.

Remarque : Le domaine considéré est obligatoirement rectangulaire. Il est possible de modéliser un domaine quelconque en ajoutant à celui-ci un jeu de digue infranchissable de par leur hauteur (voir figure 3.15).

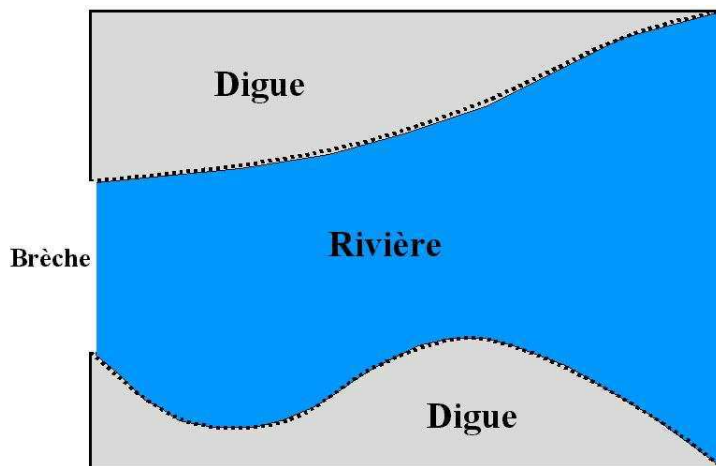


FIG. 3.15 – Construction d'un domaine non rectangulaire modélisé par LMCFLD.

3.2.7 LMCFLD en quelques chiffres

LMCFLD est un code de calcul écrit spécialement pour être dérivé dans le but de servir d'outil dans le test des méthodes d'assimilation de données. LMCFLD est codé en FOR-

3.2. UNE MAQUETTE DE CALCUL DES ÉCOULEMENTS BIDIMENSIONNEL : LMCFLD

TRAN 77, Il compte 20 sousroutines et quelques 1000 lignes de code (sans les commentaires).

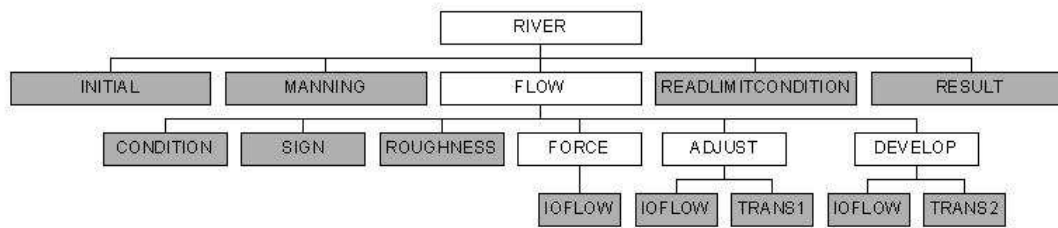


FIG. 3.16 – Organigramme du code de calcul LMCFLD

Connaître la taille du code ainsi que sa structure (cf figure 3.16) permet d'évaluer le travail à effectuer pour dériver le code de calcul.

3.2.8 Conclusion

Dans cette section, nous avons étudié le modèle d'hydraulique fluviale, LMCFLD. Basé sur la résolution des équations de Saint Venant bidimensionnelles.

Tout comme dans le cas de la modélisation 1D avec Carima, de nombreuses difficultés sont posées par la modélisation 1D.

- *Définir la topographie équivalente.* En chaque point de la grille de discrétisation, il faut définir une valeur de la topographie. Prendre la topographie réelle en chaque point de la grille, c'est ne pas tenir compte du relief situé dans chaque maille et donc risquer de ne pas calculer un écoulement réaliste.
- *Déterminer le coefficient de rugosité.* De même que pour la détermination de la topographie équivalente, la création d'un jeu de rugosité doit tenir compte de la nature du sol à l'intérieur de chaque maille.
- *Déterminer une condition aux limites* afin de tester divers scénarios. Cette condition doit être définie en chaque point d'entrée et de sortie du modèle.
- *Déterminer la condition initiale.* Il faut définir en chaque point de la grille un état initial stable. Généralement, l'état initial est obtenu après une phase de stabilisation dans la définition de la condition aux limites amont.

Les problèmes soulevés ici sont résolus manuellement par l'expérimentateur. L'assimilation de données fournit un outil de calcul automatique des différentes variables à définir. Nous verrons dans les chapitres suivants quelques exemples d'estimation de ces variables.

Chapitre 4

Mise en œuvre de l’assimilation de données : techniques et difficultés

Nous avons vu, dans le premier chapitre, la théorie liée à l’assimilation de données. Mais comme souvent la mise en œuvre de ces méthodes n’est pas forcément immédiate. Tout d’abord, afin de résoudre le système d’optimalité (2.13), il faut écrire le code adjoint correspondant. Deux voies se présentent alors, soit l’écrire directement à partir des équations continues, soit utiliser un logiciel de différentiation automatique. Nous discuterons de ce choix dans la première partie de ce chapitre.

Ensuite, une fonction coût, du type de celle définie en (2.2), doit être posée. Cette fonction est généralement celle qui permet de mesurer l’écart entre les résultats calculés par le modèle et les observations. Malheureusement, les mesures faites sur le terrain sont souvent insuffisamment nombreuses pour permettre de mener à bien une étude significative sur une vaste rivière. Mais ce manque de données ne doit pas nous empêcher de tester nos méthodes d’assimilation de données. Ainsi, pour parer à cette pénurie, nous allons avoir recours à la méthode des expériences jumelles qui sera explicitée dans la deuxième partie de ce chapitre.

Une fois l’adjoint et les observations obtenus, il ne reste plus qu’à minimiser la fonction coût. Dans ce travail de thèse, nous avons choisi d’utiliser le code de minimisation écrit par LeMarechal [12] appelé M1QN3. La troisième partie de ce chapitre sera consacrée à l’utilisation de ce code d’optimisation.

Ce chapitre présente toutes les phases à respecter afin d’obtenir les meilleurs outils pour mener à bien l’écriture du code calculant la minimisation de la fonction coût à l’aide du système adjoint. La dernière partie de ce chapitre sera consacrée à la présentation de l’algorithme de calcul de l’estimation d’un paramètre donné.

4.1 La différentiation automatique

Les méthodes d’assimilation de données procurent un outil performant aux problèmes d’identification de paramètres, d’estimation d’erreur ou d’étude de sensibilité. La mise en œuvre de ces méthodes nécessite la construction du système adjoint, c’est à dire la création d’un code permettant de modéliser le terme $\left[\frac{\partial F}{\partial X} \right]^t p$.

Pour écrire ce code dit adjoint, deux options se présentent à nous :

- soit discrétiser le système d’optimalité, puis écrire le code correspondant.
- soit réécrire un code dit adjoint à partir du code simulant le système direct.

Excepté dans le cas de l’utilisation des éléments finis non conformes, les deux approches précédentes ne sont pas équivalentes.

Par expérience, la deuxième approche donne de meilleurs résultats, mais le prix à payer pour créer ce code est très coûteux.

La dérivée du code direct peut être obtenue de deux manières différentes, soit manuellement mais cette option paraît un peu fastidieuse pour un code trop important soit automatiquement par le biais d’un logiciel de dérivation automatique.

Il existe quelques logiciels de différentiation automatique proposant divers modes de dérivation : mode tangent, dérivation d’ordre élevé, mode cotangent (adjoint) qui permettent de travailler avec différents langages : Fortran 77, Fortran 95, C ou C++.

Depuis la fin des années 90, L’INRIA (Sophia-Antipolis, projet TROPICS) développe un logiciel de différentiation automatique mettant en œuvre à la fois la dérivation en mode direct et en mode inverse : Odyssée (depuis peu une nouvelle version est accessible sur internet appelée TAPENADE). Le système Odyssée prend en entrée un code FORTRAN 77 qui calcule une fonction différentiable par morceaux et fournit en sortie un nouveau code FORTRAN 77 qui calcule la dérivée tangente ou cotangente de ce code. Le système fonctionne par transformation du code source.

Dans ce chapitre, nous discuterons des modalités d’utilisation d’Odyssée ainsi que des méthodes mises en œuvre pour la dérivation d’un code.

Bien qu’Odyssée s’améliore petit à petit en fonction des difficultés rencontrées par les utilisateurs, il reste encore beaucoup de fonctionnalités FORTRAN qui ne sont pas prises en compte. Pour écrire un code dans le but d’être différencié automatiquement, il est préférable de respecter certaines règles que nous présenterons ensuite. Le respect de ces règles permettra d’éviter les soucis rencontrés dans la dérivation des codes de calculs Carima et LMCFLD.

Un fois le code direct dérivé, il est souvent nécessaire d’intervenir manuellement dans le code adjoint afin de lever les ambiguïtés décelées par Odyssée ou bien de corriger les lignes de code superflues.

Dans la dernière partie de ce chapitre nous présenterons deux méthodes utilisées afin de vérifier l’exactitude du code adjoint et du code tangent rendu par Odyssée.

4.1.1 La théorie

Analyse du code

Odyssée dérive une branche du code direct. Afin de déterminer la sous-routine tête de cette branche, une étude minutieuse du code est nécessaire.

Tout d’abord nous devons nous poser plusieurs questions :

- Par rapport à quelles variables doit-on dériver le code direct ?
En effet, quel est l'objectif à atteindre ? Selon que l'on veuille estimer un paramètre X ou Y , nous ne dériverons pas le code en fonction des mêmes paramètres.
- Doit-on privilégier le temps de calcul ou bien la maniabilité du code adjoint ?
Une possibilité offerte par la différentiation automatique est de différencier par rapport à toutes les variables. Cette option a l'avantage de fournir un code adjoint unique quelque soit le but recherché. Mais dériver par rapport à toutes les variables allonge proportionnellement la taille du code et par conséquent le temps de calcul. Nous avons donc le choix entre un code adjoint utilisable dans toutes les conditions (cette option permet d'éviter de redériver le code direct et par conséquent d'éviter le renettoyage du code adjoint) et un code adjoint optimal en fonction du but recherché.
- Quel est la branche du code direct à dériver ?
Une fois que les variables par rapport auxquelles nous devons dériver ont été identifiées, il nous reste à déterminer quelle partie du code doit être différenciée.
Pour cela nous devons isoler la branche du code qui effectue le calcul et déterminer la sous-routine tête. L'idéal est de pouvoir découper le code direct en trois parties bien distinctes :
 - input : la partie du code qui lit les variables d'entrées **sans** qu'aucun calcul ne soit effectué,
 - calcul : c'est la branche qui nous intéresse et qui résout les équations du problème considéré (voir figure 4.1),
 - output : la partie du code qui écrit les sorties.

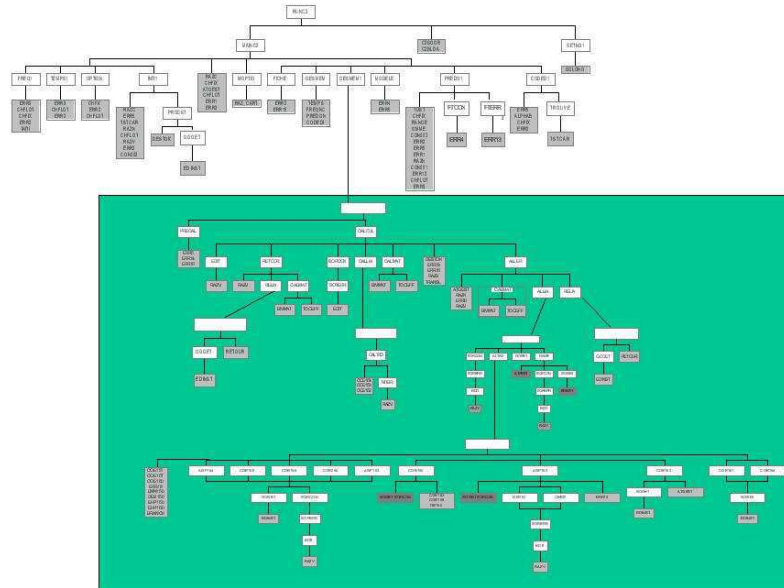


FIG. 4.1 – Organigramme du code de calcul CARIMA-SOGREAH. La partie de ce code qui doit être dérivée est mise en relief.

L'analyse du code ainsi effectuée, on va étudier maintenant les diverses fonctionnalités de Odyssee qui vont nous permettre d'obtenir le code adjoint ainsi que le code tangent.

Fonctionnalités d'Odyssée

Maintenant que nous connaissons l'ensemble des variables par rapport auxquelles nous allons dériver notre code direct ainsi que la subroutine tête, nous allons placer l'ensemble des subroutines du code direct dans un fichier unique (**Direct.f**).

Après avoir ouvert Odyssée, nous devons lire le fichier dans lequel se trouve le code direct avec la commande **load** :

load Direct

Ensuite, nous dérivons le code grâce à la commande **diff** suivie de l'option déterminant le type de dérivée désirée **-tl** pour obtenir le code tangent et **-cl** pour obtenir le code cotangent (adjoint). L'argument suivant désigne le nom de la subroutine tête (**-h**) puis vient la liste des variables par rapport auxquelles il faut dériver le code (**-vars**) :

diff -cl -h Calcul -vars i1, i2

Remarque : i1 et i2 sont des variables réelles. Effectivement, un code ne se dérive que par rapport à ses variables réelles. Les variables entières pouvant être assimilées à des fonctions constantes, leurs dérivées sont donc nulles.

Pour finir, si le système Odyssée n'a pas signalé d'erreur, il nous faut sauvegarder le code ainsi calculé à l'aide de la commande **getdiffprogram** suivie du nom donné par défaut **Directcl** puis du nom que vous voulez donner au fichier de sortie :

getdiffprogram Directcl Directcl

Le fichier **Directcl.f** ainsi obtenu contient toutes les subroutines adjointes de la branche prédéfinie. Par défaut, le nom des subroutines adjointes se termine par **cl** et le nom des variables adjointes par **ccl**.

Remarque 1 : **getinout** permet d'obtenir les variables d'entrées et de sorties de chaque subroutine. Cette commande peut donc faciliter la recherche de la liste des variables par rapport auxquelles on veut dériver la subroutine tête.

Remarque 2 : On peut de même dériver le code de façon tangente (au lieu de cotangente) en remplaçant dans la présentation précédente **cl** par **tl**.

Nous venons de voir comment il est possible d'obtenir le code tangent et le code adjoint de **Direct.f** grâce au système de différentiation automatique Odyssée. Dans le paragraphe suivant, nous allons étudier la structure du code adjoint ainsi que la manière de l'obtenir.

4.1.2 La pratique

Pour illustrer notre propos, nous allons considérer la subroutine suivante :

```
Subroutine sub1(T,N)
C
C DÉCLARATION de Variables
C
etc ...
C
C DÉBUT du code
C
etc ...
expression 1
expression 2
IF (Booleen) THEN
.      expression 3
.      expression 4
ELSE
.      expression 5
.      CALL Sub2(T,N)
.      expression 6
ENDIF
DO I = 1 , Num
.      expression 7
.      expression 8
ENDDO
expression 9
expression 10
etc ...
RETURN
END
```

Sauvegarde des trajectoires

Lorsque le système direct est non linéaire, l'évaluation du code cotangent en un point donné requiert l'usage de la trajectoire produite par le code direct. Ainsi, le code cotangent est-il construit en deux parties : le code direct dans lequel on prend soin de sauvegarder les trajectoires et le code dit "transposé" qui traduit la transposée de la jacobienne. Nous pouvons voir dans le code suivant l'illustration de l'adjoint de **sub1** :

```

Subroutine sub1CCL(T,N,TCCL)
C DÉCLARATION de Variables + adjointes
etc ...
C DÉBUT du code
C sauvegardes des trajectoires
C
etc ...
save 1
expression 1
save2
expression 2
test1 = Booleen
IF (Booleen) THEN
.   save3
.   expression 3
.   save4
.   expression 4
ELSE
.   save5
.   expression 5
.   save Sub2
.   CALL Sub2(T,N)
.   save6
.   expression 6
ENDIF
DO I = 1 , Num
.   save7
.   expression 7
.   save8
.   expression 8
ENDDO
save9
expression 9
save10
expression 10
etc ...

```

```

C Transposed part
etc ...
load10
Adjexpression 10
load9
Adjexpression 9
DO I = Num , 1 , -1
.   load8
.   Adjexpression 8
.   load7
.   Adjexpression 7
ENDDO
IF (Test1) THEN
.   load4
.   Adjexpression 4
.   load3
.   Adjexpression 3
ELSE
.   load6
.   Adjexpression 6
.   load Sub2
.   CALL Sub2CCL(T,N,TCCL)
.   load5
.   Adjexpression 5
ENDIF
load2
Adjexpression 2
load1
Adjexpression 1
etc ...
RETURN
ENDIF

```

Pour construire le code adjoint il faut réécrire le code direct avec sauvegardes de la fin vers le début, remplacer les sauvegardes par leurs lectures, puis dériver chaque expression en utilisant la méthode de transposition qui sera introduite plus tard.

Remarque : Nous pouvons voir d'ores et déjà que l'exécution du code adjoint sera forcément plus long que celui du code direct. En effet, l'exécution du code adjoint nécessite pour chaque subroutine une exécution de la subroutine directe puis une exécution de la partie transposée. De plus, comme nous le verrons dans la section suivante, chaque expression de la partie directe se démultipliera dans la partie transposée, ce qui aura pour conséquence d'augmenter encore le temps d'exécution.

Remarque : Il est possible d'éviter de sauvegarder toutes les trajectoires et ainsi de

diminuer considérablement le temps d'exécution du code adjoint. En effet, l'idée est de remplacer les sauvegarde par un recalcul de celle-ci. Cette méthode à l'avantage de diminuer la place mémoire nécessaire à l'exécution du code adjoint, mais l'inconvénient est d'alourdir le code et donc d'en assombrir la lecture.

Méthode de transposition

Dans un programme, le système direct modélisé est exprimé grâce à une succession d'instructions, ces instructions se répartissant sur plusieurs sous-routines. Le calcul du code adjoint (mathématiquement la transposée de la jacobienne) nécessite de considérer non plus le modèle mathématique mais plutôt le code informatique et donc la succession de lignes de code du code source.

Supposons que dans le code précédent **sub1**, l'expression 4 s'écrive :

$$X = Z.Y + A.X \quad (4.1)$$

où X , Y et Z sont des variables intermédiaires dépendantes des variables de dérivation et A une constante.

La ligne de code correspondante dans le code linéaire tangent s'écrira alors :

$$X' = Z'.Y + Z.Y' + A.X' \quad (4.2)$$

Maintenant, nous allons écrire cette ligne de code sous forme matricielle :

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \begin{pmatrix} A & Z & Y \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix}$$

Posons $\mathcal{X}' = (X', Y', Z')^t$ et $M = \begin{pmatrix} A & Z & Y \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$.

En introduisant le vecteur des variables adjointes $\mathcal{X}^* = (X^*, Y^*, Z^*)^t$, par définition de la transposition de la matrice M , on a :

$$(\mathcal{X}^*, \mathcal{X}') = (\mathcal{X}^*, M\mathcal{X}') = (M^t \mathcal{X}^*, \mathcal{X}')$$

qui donne

$$\mathcal{X}^* = M^t \mathcal{X}^*$$

Il vient donc :

$$\begin{pmatrix} X^* \\ Y^* \\ Z^* \end{pmatrix} = \begin{pmatrix} A & 0 & 0 \\ Z & 1 & 0 \\ Y & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X^* \\ Y^* \\ Z^* \end{pmatrix}$$

Par conséquent, dans la partie transposée du code adjoint l'expression 4 deviendra :

$$\text{AdjExpression 4} \iff \begin{cases} Z^* &= Z^* + Y.X^* \\ Y^* &= Y^* + Z.X^* \\ X^* &= A.X^* \end{cases}$$

Remarque :

- L'ordre dans lequel les instructions sont données est très important. En effet, la troisième instruction modifie la valeur de X^* , valeur qui est utilisée dans le calcul de Z^* et Y^* . Une inversion de ces instructions pourrait fausser totalement le calcul.
- Si X n'apparaît pas dans (4.1), c'est-à-dire si $A = 0.$, l'expression adjointe devient alors :

$$\text{AdjExpression 4} \iff \begin{cases} Z^* &= Z^* + Y.X^* \\ Y^* &= Y^* + Z.X^* \\ X^* &= 0.0 \end{cases}$$

Il est important de bien initialiser X^* à 0, X^* pouvant être utilisé par la suite.

- La remarque du précédent paragraphe sur le temps de calcul du code adjoint est ici bien illustrée. En effet, le temps de calcul étant déterminé principalement par le nombre de multiplications et de divisions effectuées dans le code, on voit dans cet exemple qu'une simple expression contenant deux multiplications est traduite par trois expressions contenant trois multiplications.
- Certaines fonctions ne sont pas différentiables, comme la racine carrée ou la fonction *max*. Dans ces cas bien particulier, le système Odyssee les remplace par une succession d'instructions If THEN ELSE pour pouvoir les traiter.

4.1.3 Les difficultés à contourner

La dérivation du code de calcul LMCFLD n'a posé aucun problème car il avait été écrit dans ce but, par contre la dérivation de CARIMA a été source d'énormément de soucis puisqu'il a fallu pratiquement le réécrire dans sa version simplifiée.

Ainsi dériver un code direct n'est pas toujours chose aisée. En effet, de grosses difficultés peuvent rapidement intervenir si le code n'a pas été écrit dans le but d'être dérivé. Dans ce paragraphe nous allons voir quelles sont les difficultés que l'on peut rencontrer et surtout comment les éviter.

Isoler le code de calcul

Comme nous l'avons vu précédemment, dériver un code c'est d'abord rechercher la branche de ce dernier qui contient le code de calcul. Mais dans le cas où le code de calcul n'est pas bien distinct de celui de la lecture des données d'entrée, il devient alors difficile de dériver le code. En effet les fonctions de lecture et d'écriture ne sont pas différentiables et par conséquent les sous-routines les contenant ne le sont pas non plus.

Une solution logique serait de construire son code direct en trois étapes bien distinctes :

- La lecture des données d'entrées : le stockage de ces données dans des tableaux mais aussi l'initialisation des variables. Aucun calcul ne doit être effectué dans cette partie.
- Le code de calcul.
- La sauvegarde des sorties : l'écriture des résultats dans les fichiers, etc ...

Dans ce cas de figure, il suffira de dériver le code direct en donnant la sous-routine calcul comme sous-routine de tête.

Éviter d'utiliser certaines fonctions

Dans les années 70-80, les programmeurs ont utilisé et abusé des fonctions GOTO ou assimilées (entry, cycle, etc ...). Comme nous l'avons vu, le code adjoint consiste à lire le code direct de la fin vers le début, c'est-à-dire que l'on doit pouvoir remonter l'arbre de l'exécution du code direct (voir figure 4.2). Ceci implique que les fonctions GOTO doivent être évitées puisqu'elles ne permettent pas de retrouver le chemin inverse.

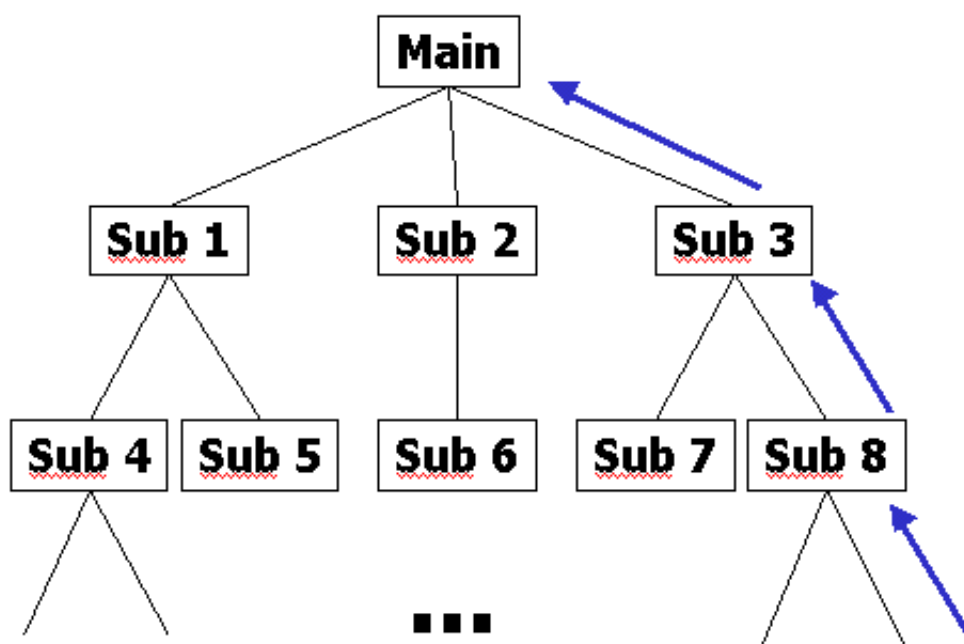


FIG. 4.2 – Arborescence classique d'un code de calcul. L'écriture du code adjoint requiert de savoir comment parcourir le code direct dans le sens de la remontée.

Déclarer le type de toutes les variables

Écrire un code adjoint, c'est aussi sauvegarder les trajectoires du code direct, c'est-à-dire qu'Odyssée crée une variable **save** à laquelle il donne le type de la variable à sauvegarder. Si le code direct n'a pas pris soin de définir un type, le différentiateur automatique va en donner un à **save** par défaut. Pour éviter toutes confusions à ce sujet il est préférable de coder le direct en utilisant la fonction **implicit none** qui oblige l'utilisateur à donner un type à toutes les variables utilisées.

Ne pas faire de sauvegardes temporaires

Pour des problèmes de capacité d'ordinateur, le recours aux fichiers de sauvegarde tampon est assez courant dans les codes d'un certain âge. Les fonctions write et read n'étant pas différentiables, il est préférable d'éviter ce type d'astuces d'autant que les machines actuelles ont assez de mémoire.

4.1.4 Modification manuelle du code d'Odyssée

Le code adjoint rendu par le système Odyssée nécessite parfois l'intervention humaine.

Pour lever les ambiguïtés décelées par Odyssée

- Supposons une boucle de taille indéterminée :

```
Entrée = True
DO While Entrée
X=Y
IF ( X .EQ. 2 ) Entrée = False
ENDDO
```

Lorsque Odyssée va sauvegarder les trajectoires dans cette subroutine, il va falloir créer un tableau SAVE pour sauvegarder la trajectoire de X dans cette boucle. Or, la taille de cette boucle est *a priori* indéterminée. Par conséquent on ne peut connaître la taille du tableau save. Par défaut, le système Odyssée va donner au tableau save la taille Odyseemax qui sera initialisée à -1 .

```
PARAMETER ( ODYSSEEMAX = -1 )
REAL SAVE(ODYSSEEMAX)
etc ...
Entrée = True
NCOUN = 1
DO While Entrée
SAVE(NCOUN) = X
X=Y
IF ( X .EQ. 2 ) Entrée = False
NCOUN = NCOUN + 1
ENDDO
etc ...
```

L'utilisateur du code adjoint devra définir lui-même une taille de tableau adéquate en fonction d'une exécution du code direct.

- Dans le cas où la fonction "implicit none" n'aurait pas été utilisée, une vérification de tous les types des sauvegardes est alors nécessaire.

Pour optimiser le code adjoint

- Toutes les sauvegardes effectuées par Odyssée ne sont pas toujours utiles. En effet certaines peuvent être enlevées sans que cela n'affecte la fiabilité du code adjoint. Soit l'exemple suivant :

SUBROUTINE SUBCCL(X,Z,XCCL,ZCCL)	
C DÉCLARATION DE VARIABLES	
REAL Y	
REAL YCCL	
REAL SAVEX	C Transposed part
REAL SAVEY	Z = SAVEZ
REAL SAVEZ	YCCL = YCCL + ZCCL
C DÉBUT DU CODE	ZCCL = 0.
SAVEY = Y	X = SAVEX
Y = X	ZCCL = ZCCL + 2.XCCL
SAVEX = X	XCCL = 0.
X = 2.Z	Y = SAVEY
SAVEZ = Z	XCCL = XCCL + YCCL
Z = Y	YCCL = 0.
END	END

Considérons SAVEX la sauvegarde de la variable X. Dans la partie transposée, la variable X n'est jamais réutilisée, il est donc inutile de sauvegarder sa trajectoire. Il en est de même pour SAVEY et SAVEZ. Le code peut alors se réécrire sans les sauvegardes tout en restant équivalent en terme de calcul de l'adjoint. Ce nettoyage des sauvegardes inutiles du code permet de gagner en rapidité d'exécution. Ci-après le code privé de ses sauvegardes inutiles :

```

SUBROUTINE SUBCCL(X,Z,XCCL,ZCCL)
C DÉCLARATION DE VARIABLES
REAL Y
REAL YCCL
C DÉBUT DU CODE
Y = X
X = 2.Z
Z = Y
END
C Transposed part
YCCL = YCCL + ZCCL
ZCCL = 0.
ZCCL = ZCCL + 2.XCCL
XCCL = 0.
XCCL = XCCL + YCCL
YCCL = 0.
END

```

- Avant d'entrer dans l'appel d'une sous-routine, le système Odyssée sauvegarde toutes les trajectoires des variables modifiées dans cette sous-routine (et dans celles appelées par cette dernière et ce de façon récursive). Pour un tableau, il suffit qu'une case de celui-ci soit modifiée pour qu'Odyssée sauvegarde le tableau entier. Il paraît judicieux de limiter la sauvegarde de ces tableaux à la seule case modifiée afin de gagner encore en rapidité.
- Si le code direct est très long, le nombre de sauvegardes effectuées dans le code adjoint associé est très important. L'exécutable ainsi généré peut alors être trop

important pour être exécuté sur une machine. Une solution simple et immédiate sera de faire les sauvegardes les plus importantes dans un fichier tampon. Cette petite astuce permet souvent de diminuer considérablement la taille de l'exécutable. Mais utiliser des fichiers temporaires de sauvegarde nécessite de connaître le type exact de la variable sauvegardée, ceci rejoignant un des conseils donné précédemment, à savoir qu'il faut déclarer dans le code direct tous les types des variables utilisées (utilisation de "implicit none" conseillée).

4.1.5 Validation du code Adjoint d'Odyssée

Pour valider les codes rendus par Odyssée, on utilise classiquement le test du gradient pour le code tangent et le test du produit scalaire pour le code cotangent.

Le test du gradient

Soit P un programme informatique tel que

$$\begin{aligned} P &: \mathbf{R}^n \rightarrow \mathbf{R}^m \\ u &\rightarrow P(u) = v \end{aligned}$$

Pour valider les procédures de dérivation de notre programme, on choisit une fonctionnelle F :

$$\begin{aligned} F &: \mathbf{R}^m \rightarrow \mathbf{R} \\ v &\rightarrow F(v) \end{aligned}$$

F peut être par exemple la norme de v .

Point de vue mathématique

Pour vérifier l'exactitude du vecteur gradient de la fonctionnelle F , considérons un développement de Taylor de F au point v .

En notant $h \in \mathbf{R}^m$ une perturbation dans la direction $\delta v \in \mathbf{R}^m$, on a

$$F(v + h) = F(v) + \nabla F(v).h + o(h)$$

Si, $\forall \delta v \in \mathbf{R}^m$, on vérifie

$$\lim_{\alpha \rightarrow 0} \frac{F(v + \alpha \delta v) - F(v)}{\alpha (\nabla F(v), \delta v)} = 1$$

alors le test du gradient est satisfait.

Algorithmique

Soit un code direct formé par un ensemble de N expressions : P , et PTL son code tangent.

Soit U et V les vecteurs d'entrée et de sortie de P .

Pour valider le code tangent on emploie la procédure suivante :

- on exécute le code direct : $V_0 = P(U)$,
- on calcule $F_0 = F(V_0)$,
- on calcule le gradient de la fonctionnelle ∇F_0 grâce au code tangent,
- pour $I = 1$, Max (Max est à définir) :

- posons $\alpha = \frac{1}{2^I}$,
- on exécute le code direct : $V_\alpha = P(U + \alpha\delta V)$,
- on calcule $F_\alpha = F(V_\alpha)$,
- puis on évalue $Limite(I) = \frac{F_\alpha - F_0}{\alpha(\nabla F_0, \delta V)}$.

Si la limite évaluée dans le dernier point tend vers 1 lorsque I est grand, alors le test du gradient est satisfait dans le direction δV .

Après avoir dérivé le code de calcul LMCFLD et obtenu son code tangent grâce à Odyssée, nous avons appliqué le test du gradient à ce dernier et obtenu le tableau suivant :

I	$1/2^I$	Limite(I)
1	0.5000000000000000	0.228477474491498
2	0.2500000000000000	0.246546338979135
3	0.1250000000000000	0.281779974799585
4	6.250000000E-002	0.343169091489788
5	3.125000000E-002	0.423112791177651
6	1.562500000E-002	0.569723825869313
7	7.812500000E-003	0.586814799088477
8	3.906250000E-003	0.694161961402958
9	1.953125000E-003	0.792815516167524
10	9.765625000E-004	0.871720956073847
11	4.882812500E-004	0.926801489931953
12	2.441406250E-004	0.961305071668677
13	1.220703125E-004	0.981782313600244
14	6.103515625E-005	0.999964963913040
15	3.051757812E-005	1.00004389713139
16	1.525878906E-005	1.02029123386864
17	7.629394531E-006	1.05385544429452
18	3.814697265E-006	1.13839260172641
19	1.907348632E-006	1.46186407549664
20	9.536743164E-007	1.99068273377142

La convergence est obtenue vers $I = 14$ où l'on obtient une valeur du gradient égale à 1 à 10^{-4} près.

Le test du produit scalaire

Le test du produit scalaire consiste en une vérification du code cotangent grâce au code tangent qui vient d'être vérifié.

Point de vue mathématique

Soit un code P défini comme précédemment :

$$\begin{aligned} P &: \mathbf{R}^n \rightarrow \mathbf{R}^m \\ u &\rightarrow P(u) = v \end{aligned}$$

Soit $P'(u)$ le code linéaire tangent calculé en un point u donné. On choisit une direction δu

de u , puis on calcule l'approximation au premier ordre δv correspondant à v par la relation

$$\delta v = P'(\delta u)$$

Soit P^* le code cotangent de P , on calcule δu^* à partir de δv comme suit :

$$\delta u^* = P^*(\delta v)$$

Si l'égalité suivante est satisfaite pour toute direction δu de u

$$(\delta v, P'(\delta u)) = (P^*(\delta v), \delta u)$$

alors le code linéaire cotangent est validé.

Algorithmique

Soit un code direct formé par un ensemble de N expressions : P , PTL son code tangent et PCL son code cotangent.

Soit U et V , les vecteurs d'entrée et de sortie de P .

Pour valider le code cotangent on emploie la procédure suivante :

- choisir une perturbation $UTTL$,
- calculer $VTTL = P'(U)$,
- initialiser $VCCL = VTTL$,
- calculer $UCCL = P^*(V)$,
- calculer le produit scalaire $\langle VTTL, VCCL \rangle = \text{PROD1}$,
- calculer le produit scalaire $\langle UTTL, UCCL \rangle = \text{PROD2}$,
- comparer $\langle UTTL, UCCL \rangle$ et $\langle VTTL, VCCL \rangle$ qui doivent être égaux.

Si les deux produits scalaires sont semblable alors le test du produit scalaire est satisfait.

Après avoir dérivé le code de calcul LMCFLD et obtenu son code adjoint grâce à Odyssée, nous avons appliqué le test du produit scalaire à ce dernier et obtenu le tableau suivant :

PROD1	1934.97507516900
PROD2	1934.97507591433

Les deux produits scalaires sont égaux à 10^{-7} près, ce qui signifie que le test du produit scalaire est bien satisfait.

4.1.6 Le second ordre

Pour mener à bien une analyse de sensibilité, nous avons vu précédemment que nous devons écrire un code adjoint au second ordre. Bien que pour l'instant Odyssée ne calcule pas directement le code adjoint au second ordre, nous allons utiliser une astuce qui permet de faire une grande partie du travail avec un différentiateur de code automatique.

Afin de calculer la sensibilité, nous devons résoudre le système (2.80), que nous rappelons ci-après :

$$\frac{dQ}{dt} + \left[\frac{\partial F}{\partial X} \right]^t Q - \left[\frac{\partial^2 F}{\partial X^2} R \right]^t P + C^t C R = 0 \quad (4.3)$$

$$\frac{dR}{dt} - \left[\frac{\partial F}{\partial X} \right] R = 0 \quad (4.4)$$

$$Q(T) = 0 \quad (4.5)$$

Le terme $\left[\frac{\partial F}{\partial X} \right]^t Q$ sera calculé grâce au code adjoint ; le terme $\left[\frac{\partial F}{\partial X} \right] R$ lui sera calculé grâce au code tangent. Il nous reste donc à créer le code qui permettra de calculer le terme $\left[\frac{\partial^2 F}{\partial X^2} R \right]^t P$.

Deux choix s'offrent à nous :

- Soit calculer le code adjoint du code adjoint. Pour cela, considérons la subroutine suivante :

```
Subroutine sub1(T,N)
C DÉCLARATION de Variables
etc ...
C DÉBUT du code
expression 1
expression 2
expression 3
etc ...
RETURN
```

Puis calculons son adjoint à l'aide d'Odyssée :

```
Subroutine sub1CCL(T,N,TCCL)
C DÉCLARATION de Variables
etc ...
C DÉBUT du code
save 1
expression 1
save 2
expression 2
save 3
expression 3
etc ...
C Transposed part
etc ...
load 3
Adjexpression 3
load 2
Adjexpression 2
load 1
Adjexpression 1
RETURN
```

Maintenant, il faut débarrasser ce code adjoint de la partie sauvegarde des trajectoires pour ne garder que la partie transposée, ce qui donne le code suivant :

```

Subroutine sub1CCL(T,N,TCCL)
C DÉCLARATION de Variables
etc ...
C DÉBUT du code
etc ...
Adjexpression 3
Adjexpression 2
Adjexpression 1
RETURN

```

Il faut maintenant calculer le code adjoint de cette subroutine grâce à Odyssée. Le code obtenu est donc :

```

Subroutine sub1CCLCCL(T,N,TCCL,TCCLCCL)
C DÉCLARATION de Variables
etc ...
C DÉBUT du code
etc ...
save 3
Adjexpression 3
save 2
Adjexpression 2
save 1
Adjexpression 1
C Transposed part
load 1
AdjAdjexpression 1
load 2
AdjAdjexpression 2
load 3
AdjAdjexpression 3
etc ...
RETURN

```

Maintenant, il suffit de rajouter à ce code la partie sauvegarde des trajectoires effacées précédemment afin de respecter toutes les initialisations de variables (les variables directes ainsi que les variables adjointes du premier ordre). Le code adjoint au second ordre est ainsi obtenu :

```

Subroutine sub1CCLCCL(T,N,TCCL,TCCLCCL)
C DÉCLARATION de Variables
etc ...
C DÉBUT du code
save 1
expression 1
save 2
expression 2
save 3
expression 3
etc ...
C Transposed part
etc ...
load 3
save 33
Adjexpression 3
load 2
save 22
Adjexpression 2
load 1
save 11
Adjexpression 1

```

```

C Transposed part second
load 11
AdjAdjexpression 1
load 22
AdjAdjexpression 2
load 33
AdjAdjexpression 3
etc ...
RETURN

```

- Soit calculer le code adjoint du code tangent. Pour cela considérons la subroutine suivante :

```

Subroutine sub1(T,N)
C DÉCLARATION de Variables
etc ...
C DÉBUT du code
expression 1
expression 2
expression 3
etc ...
RETURN

```

Puis calculons son tangent à l'aide d'Odyssée :

```

Subroutine sub1TTL(T,N,TTTL)
C DÉCLARATION de Variables
etc ...
C DÉBUT du code
Tgtexpression 1
Tgtexpression 2
Tgtexpression 3
etc ...
RETURN

```

Il faut maintenant calculer le code adjoint de cette subroutine grâce à Odyssée, d'où :


```

Subroutine sub1TTLCCL(T,N,TTL,TTTLCCL)
C DÉCLARATION de Variables
etc ...
C DÉBUT du code
save 1
Tgtexpression 1
save 2
Tgtexpression 2
save 3
Tgtexpression 3
etc ...
C Transposed part
etc ...
load 3
AdjTgtexpression 3
load 2
AdjTgtexpression 2
load 1
AdjTgtexpression 1
RETURN

```

Les deux approches devraient être équivalentes. Cependant, calculer l'adjoint du code adjoint, génère un code qui nécessite une plus grande capacité machine que la deuxième option. En effet, le nombre de sauvegarde à effectuer est exactement le double. Ainsi, pour des codes de grandes tailles qui nécessitent déjà beaucoup de mémoire, la capacité deviendrait ingérable. Nous allons donc opter pour la deuxième option qui est de choisir de prendre l'adjoint du tangent.

4.1.7 Conclusion

Nous avons vu précédemment les principales fonctionnalités du système de différentiation automatique Odyssée. Ce type d'outil performant permet de calculer rapidement l'adjoint d'un code donné. Il est évident que, pour les codes assez longs, le temps économisé par rapport à une dérivation manuelle est loin d'être négligeable.

Toutefois, ce travail nécessite une connaissance approfondie du code et une approche rigoureuse.

Depuis le début de ce travail de doctorat, le système Odyssée a évolué vers un nouveau système TAPENADE. Ce dernier est accessible via le web (<http://tapenade.inria.fr:8080/tapenade/index.jsp>) et traite un plus grand nombre de fonction Fortran.

Bien que certains points abordés dans ce chapitre sont maintenant résolus par TAPENADE, la marche à suivre pour dériver un code reste valable.

4.2 Expériences jumelles

L'assimilation de données est une technique mathématiques de contrôle optimal qui nécessite la minimisation d'une fonctionnelle mesurant la différence entre les observations faites sur le domaine étudié et les résultats calculés par le modèle. En hydraulique fluviale, les observations sont très rares en période d'inondation. Ceci est dû à la manière de faire

ces mesures. En effet, la plupart du temps ces mesures sont effectuées à l'aide de capteurs placés sur la zone. Ces derniers peu nombreux, sont de préférence positionnés dans le lit de la rivière. Ils permettent donc d'observer l'évolution de la crue dans le lit mineur, lorsqu'ils ne sont pas emportés par la violence des flots. Les photos satellites (ou prises d'avion) n'étant pas très répandues à cause de la rapidité des événements, les seules observations disponibles sont alors les marques que l'eau a pu laisser sur les murs des maisons. Le problème de ces données est qu'elles ne peuvent pas être datées dans le temps.

Afin de pallier ce manque de données, nous allons utiliser une technique, dite des expériences jumelles, qui permet de créer artificiellement des observations à partir d'une exécution du code direct.

Cette méthode ne faisant pas intervenir d'observations réelles, elle n'a pour seul but que de valider les méthodes testées.

Mise en œuvre

Soit K un jeu de données que nous cherchons à estimer grâce à la résolution du système d'optimalité (paramètres empiriques ou non, conditions aux limites ou initiale).

Soit P le code de modélisation hydraulique.

Une exécution du code direct P va nous permettre d'obtenir les vitesses et les hauteurs d'eau sur le domaine étudié. Soit Res ces résultats :

$$P(K) = Res \quad (4.6)$$

Nous allons maintenant considérer que ces résultats Res vont être notre jeu d'observations.

A partir de δK une perturbation du jeu de paramètres K , le code direct P nous donne un nouveau jeu de résultats Res' :

$$P(K + \delta K) = Res' \quad (4.7)$$

Nous allons donc utiliser le jeu d'observations Res et le jeu de paramètres $K + \delta K$ afin d'essayer de retrouver le jeu de paramètres K grâce à l'utilisation du code adjoint et à la minimisation de la fonctionnelle suivante :

$$J(K) = \frac{1}{2} \|CRes'(K) - Res\|_{\chi_{obs}}^2 \quad (4.8)$$

Cette manière classique d'obtenir un jeu de données a le double avantage de fournir autant d'observations que possible afin de les utiliser comme bon nous semble mais aussi d'éviter les erreurs dues au modèle.

4.3 Le code de minimisation M1QN3

4.3.1 Description de la méthode

M1QN3 est un code de minimisation sans contrainte, c'est-à-dire qu'il résout le problème suivant :

$$\min (f(x) , x \in \mathbf{R}^n) \quad (4.9)$$

A chaque itération, M1QN3 calcule la direction de descente d_k de f au point x_k grâce à une formule de type :

$$d_k = -H_k \cdot g_k \quad (4.10)$$

où H_k est la discrétisation du Hessian $\nabla^2 f(x)$ de f au point x_k et g_k est le gradient de f au point x_k .

Puis le pas de descente ρ_k suivant la direction d_k est calculé grâce à une procédure de recherche linéaire.

Enfin, l'itération suivante est calculée avec la formule classique suivante :

$$x_{k+1} = x_k + \rho_k \cdot d_k \quad (4.11)$$

Pour plus de renseignements sur la méthode quasi Newton employée, on se référera à l'article de LeMarechal [12].

4.3.2 Utilisation du code

L'utilisation de M1QN3 est très aisée mais nécessite tout de même une mise en œuvre rigoureuse.

Les arguments de la subroutine M1QN3 sont définis successivement comme suit :

- SIMUL : le nom de la subroutine qui fait appel au code adjoint du modèle d'hydraulique. Cette subroutine doit être déclarée en EXTERNAL dans le programme qui appelle M1QN3.
- PROSCA : le nom de la subroutine qui calcule le produit scalaire. Cette subroutine doit être déclarée en EXTERNAL dans le programme qui appelle M1QN3.
- NDIM : dimension du vecteur de paramètres.
- X : vecteur de paramètres.
- FONCTION : valeur de la fonction coût au point X .
- GRADX : gradient de la fonction coût au point X .
- DXMIN : résolution de X . Tout point distant de moins de DXMIN sera considéré comme identique.
- DF1 : estimation de la pente de la fonction coût.
- EPSG : critère de sortie du code d'optimisation.
- IMPRES, IO, MODE : critère spécifique au code M1QN3.
- NITER : nombre d'itération maximale acceptée.
- NSIM : nombre d'appel à la subroutine SIMUL maximal.
- RZ, NRZ, IZS, RZS, DZS : tableaux permettant de passer en argument ce que désire l'utilisateur.

Avant de faire appel à la subroutine M1QN3, nous devons initialiser l'algorithme de minimisation grâce à l'appel de la subroutine SIMUL. Cette dernière permet de calculer le gradient de la fonction coût en faisant appel au code adjoint.

Les arguments de cette subroutine sont successivement définis comme suit :

- INDIC : indice d'entrée dans la subroutine SIMUL.
- NDIM : dimension du vecteur de paramètres.
- X : vecteur de paramètres.
- FONCTION : valeur de la fonction coût au point X .
- GRADX : gradient de la fonction coût au point X .

- IZS, RZS, DZS : tableaux permettant de passer en argument ce que désire l'utilisateur.

La démarche précédente ainsi respectée permettra une utilisation optimale de ce code de minimisation.

4.4 Algorithme pour l'estimation du paramètre K

Les premières sections de ce chapitre nous ont appris à obtenir et à utiliser les différents outils permettant la création des codes d'estimation de paramètres.

Nous allons dérouler l'algorithme permettant l'estimation d'un paramètre K (voir la section 2.1.4) grâce à la minimisation d'une fonction coût.

Soit le code direct **Direct.f**. Considérons que ce code est organisé comme suit :

Read la première partie de Direct.f est allouée à la lecture des variables d'entrées afin d'initialiser le calcul.

Calcul la deuxième partie est celle du calcul.

Write cette dernière partie est consacrée à l'écriture des sorties.

C'est donc la partie calcul qui devra être dérivée afin d'obtenir le code adjoint ; appelons cette partie **calcul.f**.

4.4.1 Algorithme de la subroutine principale

Initialisation. Tout d'abord il faut initialiser les variables locales.

Lecture des entrées. Dans un premier temps il faut initialiser le code de calcul grâce à l'appel de la subroutine Read.

Sauvegarde de l'état initial. On sauvegarde l'état initial afin de pouvoir faire tourner le code de calcul à chaque fois dans les mêmes conditions

Prises en compte des observations. 1. Soit il faut lire les observations faites *in situ* dans un fichier.

2. Soit utiliser la méthode des expériences jumelles :

Exécution du code de calcul. On appelle calcul.f.

Sauvegarde des observations. Les résultats de l'exécution de calcul.f sont sauvegardés dans le vecteur des observations.

Initialisation du paramètre à estimer, K . On initialise de manière aléatoire le paramètre K .

Sauvegarde du paramètre K dans un vecteur X . Afin de respecter les arguments de la subroutine M1QN3 ainsi que ceux de la subroutine SIMUL, nous devons sauvegarder la variable K dans un vecteur X .

Remarque : La variable K peut être un scalaire tout comme un vecteur ou une matrice. Quel que soit le type de K , il faudra le sauvegarder dans un vecteur X .

$$X \leftarrow K$$

Appel de SIMUL. Afin d'initialiser certaines variables nécessaires à la subroutine M1QN3, il faut faire appel au code SIMUL. Nous verrons dans la suite quel doit être l'algorithme de SIMUL.

Initialisation des variables de M1QN3. Avant d'appeler M1QN3, nous devons initialiser certaines variables nécessaires à son bon déroulement, comme le nombre d'itération maximal ou le critère d'arrêt.

Appel de M1QN3. Enfin on fait appel au code M1QN3 qui va minimiser la fonction coût.

Récupération du paramètre optimal. Après l'exécution de M1QN3, il suffit de récupérer le paramètre optimal calculé dans le vecteur X .

Il nous reste à définir l'algorithme de la subroutine SIMUL.

4.4.2 Algorithme de la subroutine SIMUL

Lecture de l'état initial. On commence par initialiser l'exécution du code de calcul.

Initialisation des variables adjointes. On initialise les variables adjointes à 0.

Lecture du paramètre à estimer. On lit dans le vecteur X la valeur actuelle du paramètre K .

$$K \Leftarrow X$$

Initialisation de la valeur adjointe du paramètre K . On initialise l'exécution du code adjoint. L'initialisation de la variable adjointe p dans le système d'optimalité est $p(T) = 0$. Nous allons donc initialiser $KCCL$ à 0.

$$KCCL = 0. \Leftrightarrow p = 0.$$

DO $t = T-1$, 0 la boucle temporelle

Appel du code adjoint FluxAdj. On appelle ici le code adjoint d'une subroutine Flux que nous introduirons plus tard.

Calcul de la variable adjointe p . On calcule maintenant la variable adjointe à l'itération suivante grâce à la discrétisation du système adjoint :

$$P = P - KCCL + C^t (C.X - X_{obs})$$

ENDDO

Initialisation de GRADX. Enfin on initialise la variable GRADX nécessaire à l'exécution du code M1QN3.

$$GradX \Leftarrow P$$

Enfin, il nous reste à définir la subroutine Flux.

4.4.3 Algorithme de la subroutine Flux

Cette subroutine va nous permettre de calculer successivement les résultats du direct et la valeur de la fonction coût.

Appel de calcul. On appelle d'abord la subroutine de calcul afin d'obtenir un jeu de résultats pour calculer la valeur de la fonction coût.

Calcul de la fonction coût. On calcule ensuite la valeur de la fonction grâce aux résultats obtenus précédemment ainsi qu'au jeu d'observations.

Le code adjoint de Flux est utilisé dans la subroutine SIMUL. Il permet de calculer la variable adjointe du paramètre à estimer ainsi que la valeur de la fonction coût nécessaire à l'exécution du code M1QN3.

4.5 Conclusion

Ce chapitre montre la façon dont on peut obtenir le code adjoint au premier comme au deuxième ordre grâce à l'utilisation d'un logiciel de différentiation automatique. Une fois le système adjoint écrit, il ne reste plus qu'à le coupler avec le modèle de minimisation afin de mener à bien la résolution du système d'optimalité.

Avec l'aide des expériences jumelles, nous pouvons éviter les problèmes associés au manque d'observations.

Le travail de mise en œuvre des techniques d'assimilation de données nécessite une grande rigueur et un code direct parfaitement écrit.

Chapitre 5

Assimilation de données et prévision

La connaissance de la condition initiale est primordiale. Une prévision fiable se fonde forcément sur des bases solides, à savoir une condition initiale optimale. Supposons que l'évolution d'un système (hydraulique, météorologique ou autres) puisse se représenter par une trajectoire dans le temps. Idéalement, la prévision calculée par la résolution des équations régissant ce système doit suivre méticuleusement cette trajectoire. Ceci n'est pas possible si une erreur est commise dès l'état initial. La condition initiale doit donc être estimée au mieux afin d'éviter tous problèmes d'initialisation.

Dans le domaine de la prévision, les conditions aux limites imposées en amont et en aval d'une rivière sont tout aussi importantes. En effet, afin de mener à bien la modélisation d'une rivière, il est capital de bien connaître l'écoulement entrant ainsi que l'écoulement correspondant en aval.

Dans ce chapitre, nous allons voir de nombreux exemples sur l'utilisation du modèle adjoint pour l'estimation de la condition initiale et des conditions aux limites. Ce travail sera mené avec un logiciel de modélisation hydraulique monodimensionnel, Carima, et un logiciel d'hydraulique fluviale bidimensionnel, LMCFLD. Le but de ce chapitre est de montrer l'intérêt des méthodes d'assimilation de données pour l'estimation des conditions initiales et aux limites.

Les données utilisées pour la construction des exemples proviennent de deux sources distinctes.

- virtuelle, afin d'établir des cas tests intéressants qui permettent de se focaliser sur des particularités physiques dont l'influence sur l'écoulement est significative,
- réelles, afin de montrer les possibilités des méthodes d'assimilation de données dans des cas concrets.

Cependant, quel que soit l'origine des données utilisées pour la construction des modèles, aucun exemple présenté n'utilise d'observations réelles. En effet, comme il a déjà été dit dans les chapitres précédents, le manque d'observations en hydraulique est trop important pour tester les méthodes d'assimilation de données de manière optimale. Ainsi, nous ferons appel à la méthode des expériences jumelles afin d'obtenir un jeu d'observations complet (une donnée à chaque pas de temps et pour chaque point de grille).

Dans un premier temps, une illustration de l'estimation de la condition initiale sera présentée grâce à l'adjoint du modèle Carima. Nous verrons ensuite une étude de cette

même estimation avec LMCFLD. Enfin, quelques exemples sur l'estimation de la condition aux limites seront exposés.

5.1 Estimation de la condition initiale

5.1.1 Carima

La dérivation du code de calcul Carima a posé de nombreux problèmes dus à l'âge du code. En effet, certaines fonctionnalités employées dans le développement de Carima ne sont pas compatibles (ou très difficilement) avec l'écriture d'un code adjoint. Dans les années 70, les machines avaient une faible capacités de stockage qui nécessitait de connaître à l'avance la capacité mémoire requise pour faire tourner un code.

Par conséquent, les programmeurs ont créé des tableaux gigantesques (de plusieurs millions de cases) afin d'y stocker toutes les variables utilisées par le code sans se soucier du type de celles-ci. Malheureusement, comme nous l'avons vu précédemment, l'écriture du code adjoint nécessite la connaissance du type de toutes les variables.

De plus, elle requiert une sauvegarde de chaque tableau modifiés dans le direct ce qui peut générer une surcharge de la machine dans le cas où ces tableaux gigantesques seraient sauvegardés plusieurs fois. Le manipulation de ces grands tableaux ne peut donc aller de pair avec l'écriture d'un code adjoint. La création de ce nouveau code nécessite donc la réécriture presque totale de Carima.

Devant ce travail gigantesque, il a fallu faire des choix pour simplifier Carima et par conséquent pour diminuer la taille du code. On s'est donc limité aux exemples où l'écoulement est fluvial et la rivière est sans singularité, ce qui nous a permis d'éliminer toutes les lignes de code traitant de l'écoulement dans les plaines inondables et des singularités du lit mineur. La longueur du code est donc passée de 10000 lignes à près de 3400.

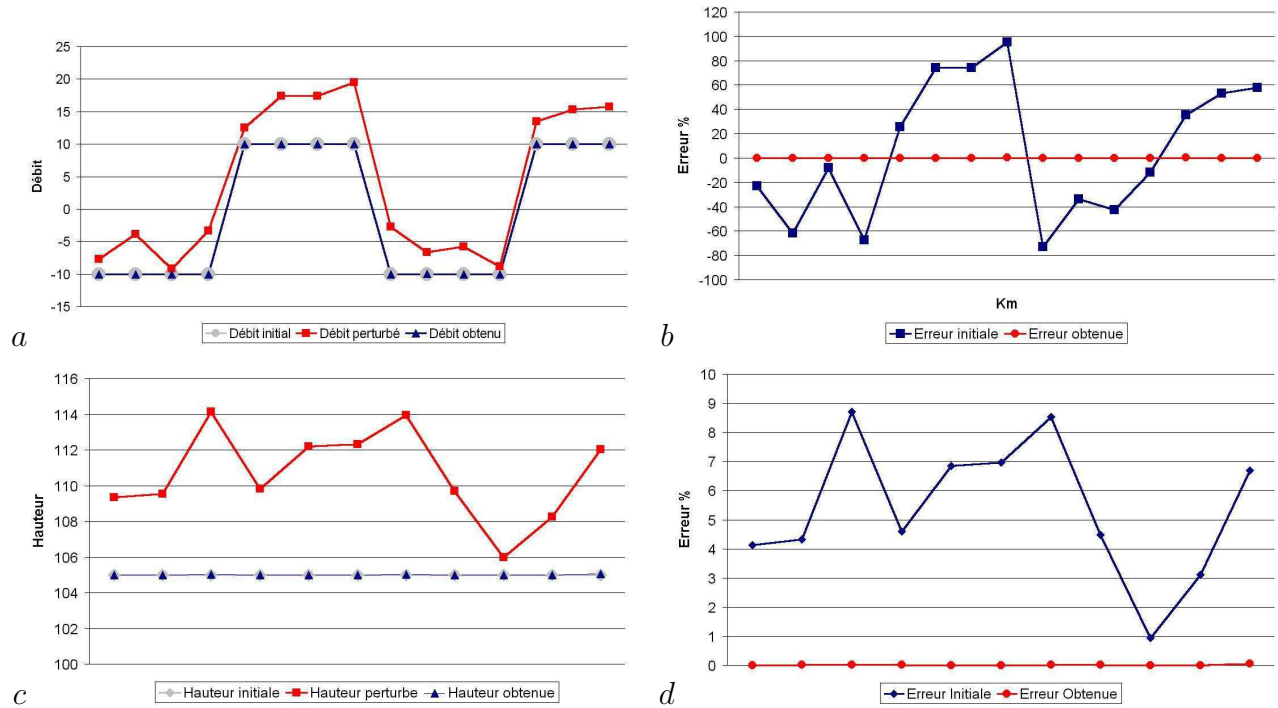
Dans la suite de cette partie nous allons aborder deux exemples sur l'estimation de la condition initiale afin de tester la méthode de l'adjoint, d'abord avec une rivière à structure maillée ensuite sur une partie de l'Isère allant de Pontcharra à Grenoble.

Une rivière maillée

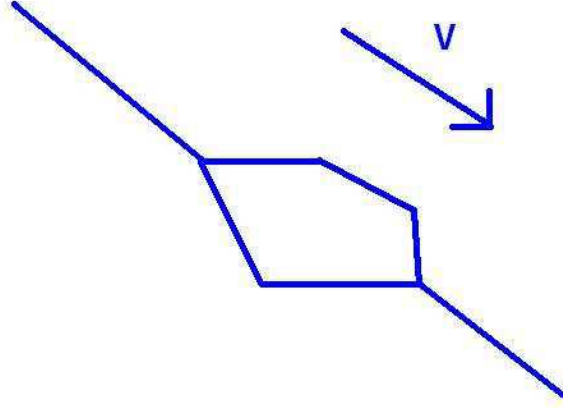
Cette rivière fictive de 8 Km de long est composée d'une première partie linéaire, suivie d'une bifurcation simulant une île (cf figure 5.1) et finit par une dernière partie linéaire. Les coupes transversales définissant cette rivière sont trapézoïdales et forment un lit régulier dont la pente est à 2%. Le coefficient de rugosité (Strickler) est égal à 15 dans tout le lit de la rivière.

Afin de tester les méthodes d'assimilation de données, nous avons fait appel à la méthode des expériences jumelles pour obtenir un jeu d'observations sur les hauteurs d'eau et les débits.

Ensuite nous devons définir une fonction coût qui devra être minimisée afin de retrouver la condition initiale optimale (cf tableau 5.1, figures a et c, courbes grises, point rond) pour



TAB. 5.1 – Illustration de l'estimation de la condition initiale pour une rivière simple. a- (resp. c-) La courbe grise (point rond) désigne la valeur de la condition initiale sur le débit (resp. la hauteur d'eau) à retrouver, la courbe rouge (point carré), la valeur perturbée de la condition initiale sur le débit (resp. la hauteur d'eau) et la courbe bleue (point triangulaire), la valeur de la condition initiale sur le débit (resp. la hauteur d'eau) obtenue par la minimisation de la fonction coût. b- (resp. d-) La courbe en bleu (point carré) est l'erreur initiale sur le débit (resp. la hauteur d'eau) et la courbe en rouge (point rond) est l'erreur finale sur le débit (resp. la hauteur d'eau). Le débit est exprimé en $\text{m}^3 \cdot \text{s}^{-1}$ et la hauteur d'eau en m.

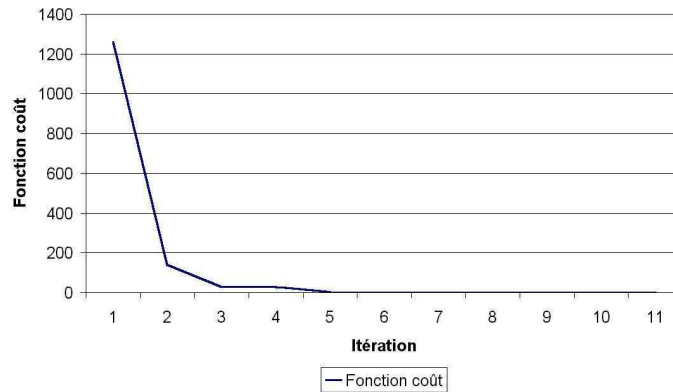
FIG. 5.1 – *Cas test de la rivière simple pour le logiciel Carima.*

le jeu d'observations précédent. Cette fonction sera définie comme suit :

$$J(CI) = \frac{1}{2} \|Q - Q_{obs}\|^2 + \frac{1}{2} \|Z - Z_{obs}\|^2 \quad (5.1)$$

où Q est le débit, Z la hauteur d'eau et X_{obs} les observations fournies par la méthode des expériences jumelles (avec $X = Q, Z$). Les observations sont disponibles au temps final.

Nous allons débiter l'exécution du code d'optimisation avec une condition initiale perturbée aléatoirement (cf tableau 5.1 figures a et c, courbes rouges, point carré) par rapport à la condition initiale optimale. Le but de la minimisation va être de retrouver une valeur de la condition initiale (cf tableau 5.1 figures a et c, courbes bleues, point triangulaire) proche de celle à l'optimum.

FIG. 5.2 – *Cas test de la rivière simple traité par le code de calcul Carima : évolution de la fonction coût en fonction de l'itération du code de minimisation.*

Après 5 itérations du code d'optimisation, le minimum de la fonction coût a été atteint (cf figure 5.2) et une condition initiale a été obtenue. Ce résultat est très proche de celui

attendu puisque l'erreur commise est en moyenne inférieure à $10^{-2}\%$ sur la longueur de la rivière (cf tableau 5.1 figures b et d, courbes rouges, point rond).

N.B : Afin de mesurer l'erreur commise et compte tenu du fait que nous connaissons exactement la variable à estimer, nous mesurons l'erreur commise grâce à la norme 2. Ainsi, c'est l'erreur relative que nous mesurons. Dans la suite de ce mémoire, c'est cette erreur qui sera toujours utilisée.

De ce premier exemple, nous pouvons déjà tirer quelques enseignements :

- la minimisation de la fonction coût est assez rapide puisque 5 itérations sont suffisantes pour converger vers une solution satisfaisante.
- La condition initiale obtenue est très proche de celle attendue. Les techniques d'assimilation de données peuvent apporter une précision intéressante.
- Le jeu d'observations utilisé dans cet exemple est très dense, c'est-à-dire que nous avons pour chaque point de calcul toutes les informations possibles et que nous les utilisons dans la définition de la fonction coût. Cette précision irréaliste est sûrement à l'origine de la petitesse de l'erreur commise. Une dégradation de ce jeu de données provoquerait sans doute une diminution de l'exactitude de la condition initiale trouvée. Nous étudierons le cas de la dégradation du jeu d'observation dans la suite de ce travail.

L'Isère

Dans cet exemple, nous allons utiliser les données réelles définissant l'Isère de Pontcharra à Grenoble (voir figure 5.3). C'est une rivière de 50 Km de long.

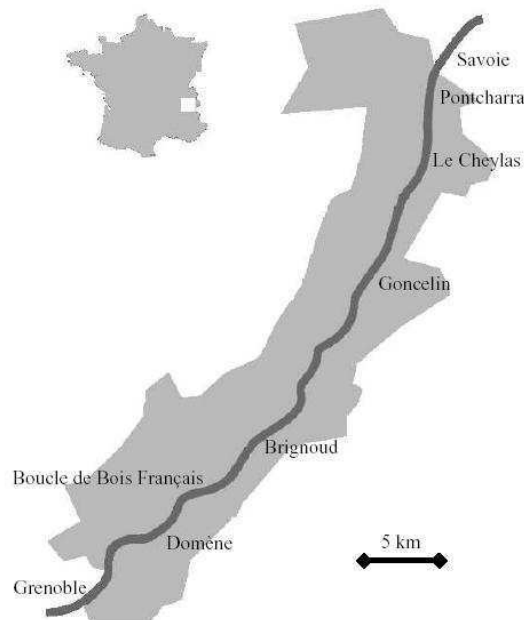
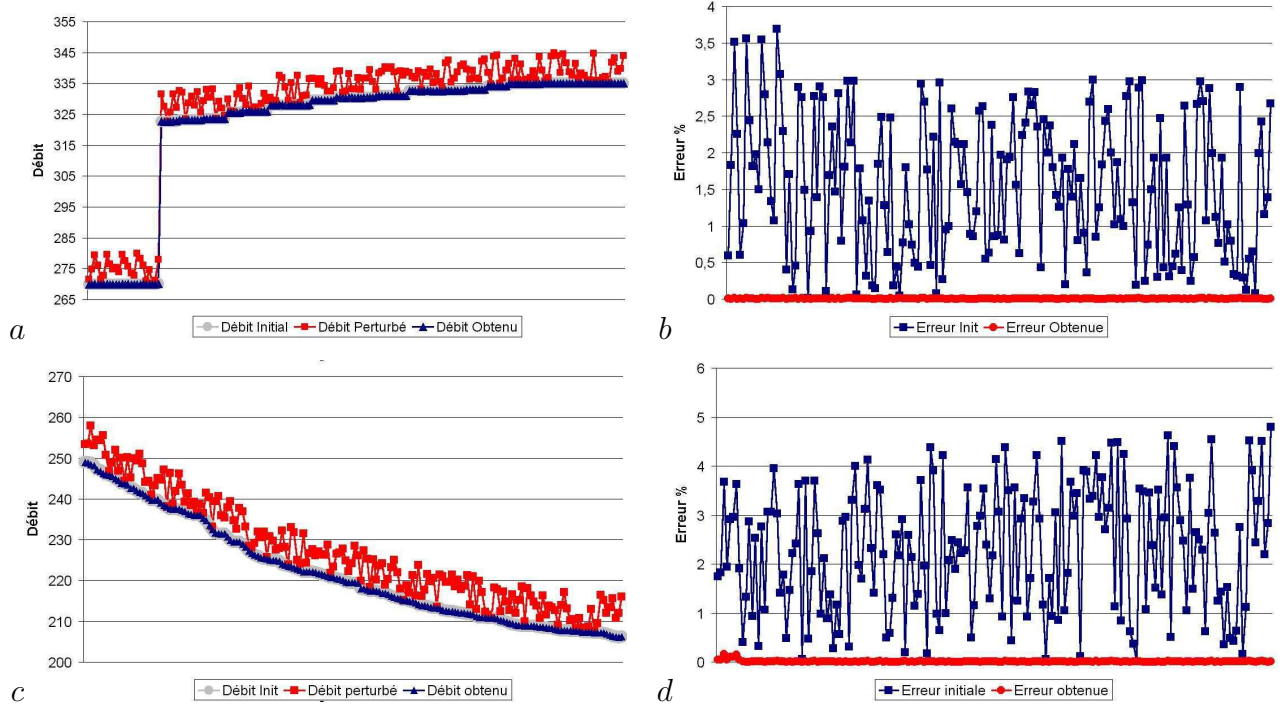


FIG. 5.3 – Représentation de l'Isère entre Pontcharra et Grenoble. Ce cas réel est utilisé dans le cadre de l'estimation de la condition initiale avec le logiciel Carima.



TAB. 5.2 – Illustration de l'estimation de la condition initiale pour le cas test de l'Isère. a- (resp. c-) La courbe grise (point rond) désigne la valeur de la condition initiale sur le débit (resp. la hauteur d'eau) à retrouver, la courbe rouge (point carré), la valeur perturbée de la condition initiale sur le débit (resp. la hauteur d'eau) et la courbe bleue (point triangulaire), la valeur de la condition initiale sur le débit (resp. la hauteur d'eau) obtenue par la minimisation de la fonction coût. b- (resp. d-) La courbe en bleu (point carré) est l'erreur initiale sur le débit (resp. la hauteur d'eau) et la courbe en rouge (point rond) est l'erreur finale sur le débit (resp. la hauteur d'eau). Le débit est exprimé en $m^3.s^{-1}$ et la hauteur d'eau en m.

Afin de tester les méthodes d’assimilation de données, nous avons fait appel à la méthode des expériences jumelles pour obtenir un jeu d’observations sur les hauteurs d’eau, les débits et les vitesses.

Comme nous l’avons fait précédemment, il faut maintenant définir une fonction coût à minimiser :

$$J(CI) = \frac{1}{2} \|Q - Q_{obs}\|^2 + \frac{1}{2} \|Z - Z_{obs}\|^2 \quad (5.2)$$

où Q est le débit, Z la hauteur d’eau et X_{obs} les observations fournies par la méthode des expériences jumelles (avec $X = Q, Z$). Les observations sont disponibles au temps final.

Nous allons débiter l’exécution du code d’optimisation avec une condition initiale perturbée aléatoirement (cf tableau 5.2 figures a et c, courbes rouges, point carré) par rapport à la condition initiale optimale. Le but de la minimisation va être de retrouver une valeur de la condition initiale (cf tableau 5.2 figures a et c, courbes bleues, point triangulaire) proche de celle à l’optimum.

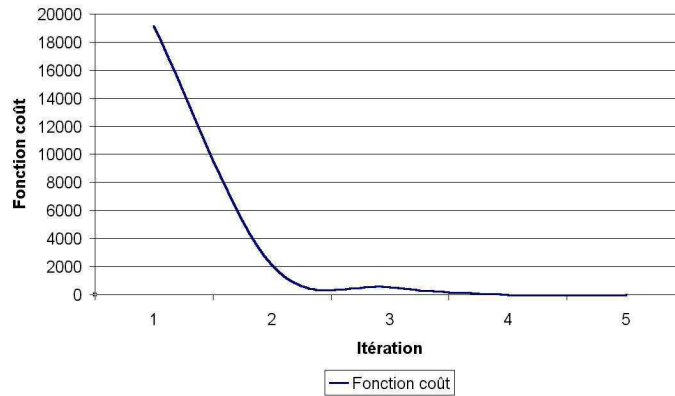


FIG. 5.4 – Cas test de l’Isère pour le code de calcul Carima : évolution de la fonction coût en fonction de l’itération du code de minimisation.

Après 4 itérations du code d’optimisation, le minimum de la fonction coût a été atteint (cf figure 5.4) et une condition initiale a été obtenue. Ce résultat est très proche de celui attendu puisque l’erreur commise est en moyenne inférieure à 0.01% sur la longueur de la rivière (cf tableau 5.2 figures b et d, courbes rouges, point rond).

Les conclusions que nous pouvons tirer de ce deuxième exemple sont identiques à celle déduites de l’exemple précédent. On peut toutefois souligner l’utilisation de données réelles pour définir la rivière.

5.1.2 LMCFLD

Le code de calcul LMCFLD a été écrit en vue d'être dérivé. L'écriture du code adjoint n'a donc posé aucun problème.

Dans cette section nous allons aborder deux exemples sur l'estimation de la condition initiale, tout d'abord avec un bassin à fond plat, ensuite nous allons introduire un peu de relief sur le fond de ce bassin avec une légère pente.

Un bassin à fond plat

Le domaine étudié ici est formé d'un bassin carré (50 m × 50 m) dont la partie ouest est bordée d'une digue infranchissable. Le fond de ce bassin est plat. L'eau se déverse par une brèche rectangulaire pratiquée dans la digue.

La condition aux limites ouest sera donc définie grâce à des couples (hauteur, temps).

Les conditions aux limites nord et sud sont de type glissement sans frottement :

$$\vec{u} \cdot \vec{n} = 0$$

et la condition aux limites est est de type frontière ouverte

$$\frac{\partial \vec{u}}{\partial \vec{n}} = 0$$

Le temps de l'écoulement sera de 1000 secondes avec un pas de temps de 2 secondes.

Afin de tester les méthodes d'assimilation de données, nous avons fait appel à la méthode des expériences jumelles pour obtenir un jeu d'observations sur les hauteurs d'eau et les vitesses.

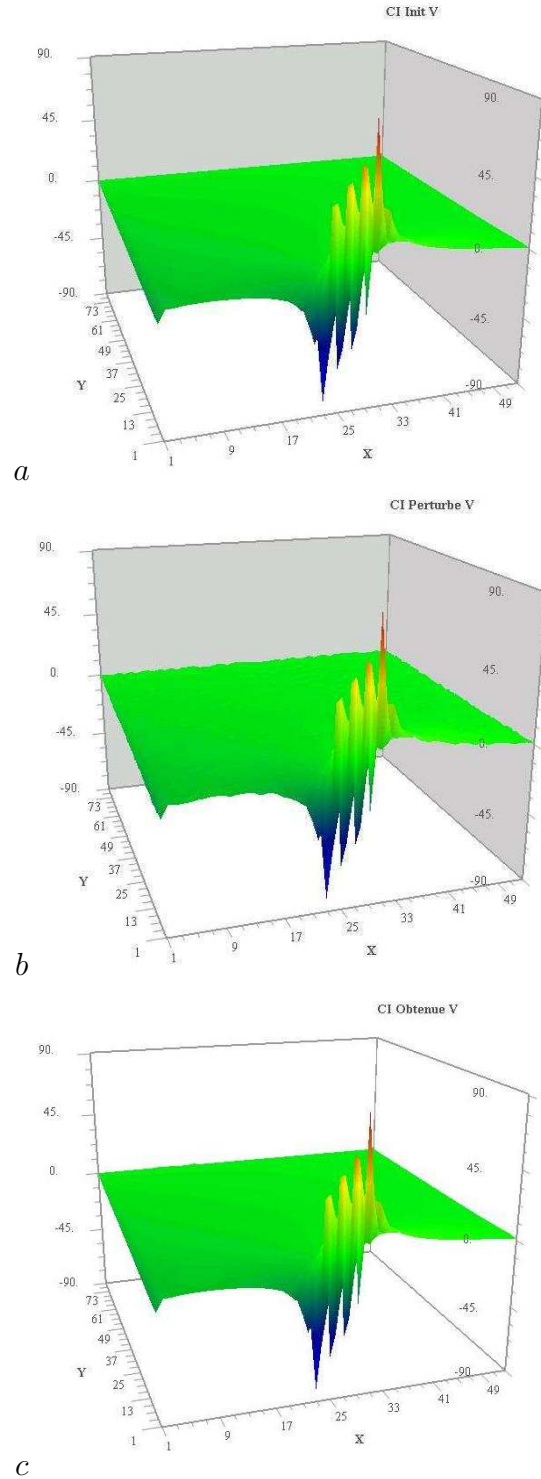
Comme nous l'avons fait précédemment, il faut maintenant définir une fonction coût à minimiser :

$$J(CI) = \frac{1}{2} \|U - U_{obs}\|^2 + \frac{1}{2} \|V - V_{obs}\|^2 + \frac{1}{2} \|Zn - Zn_{obs}\|^2 \quad (5.3)$$

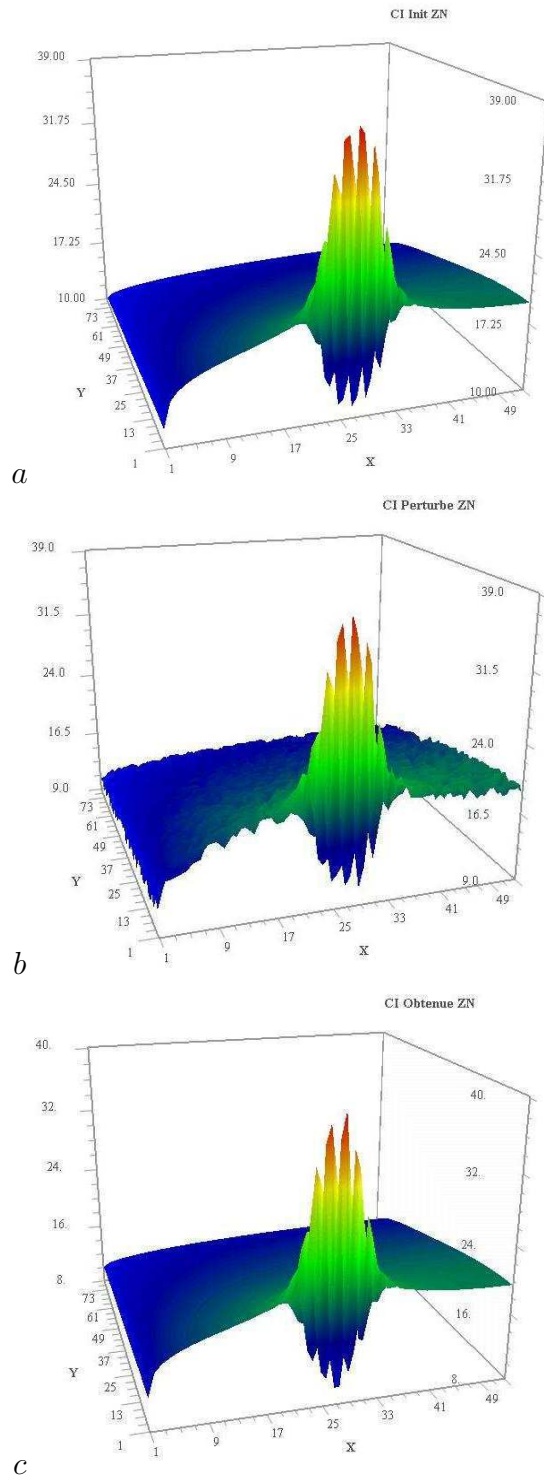
où (U, V) sont les coordonnées de la vitesse (U est l'écoulement ouest-est et V est l'écoulement nord-sud), Zn la hauteur d'eau et X_{obs} les observations fournies par la méthode des expériences jumelles (avec $X = U, V, Zn$). Les observations sont disponibles au temps final.

Avec un jeu dense d'observations et sur un domaine où le relief n'est pas chaotique, l'estimation de la condition initiale donne des résultats satisfaisants (cf figures c dans les tableaux 5.3 et 5.4), puisque sur une seule exécution on peut estimer à la fois la condition initiale donnée en vitesse et en hauteur d'eau. On peut voir sur les figures dans les tableaux précédent que bien que la condition initiale donnée pour débiter l'exécution du code d'optimisation soit assez éloignée de l'optimale, l'erreur commise en V et en Zn (cf tableau 5.5) est assez insignifiante.

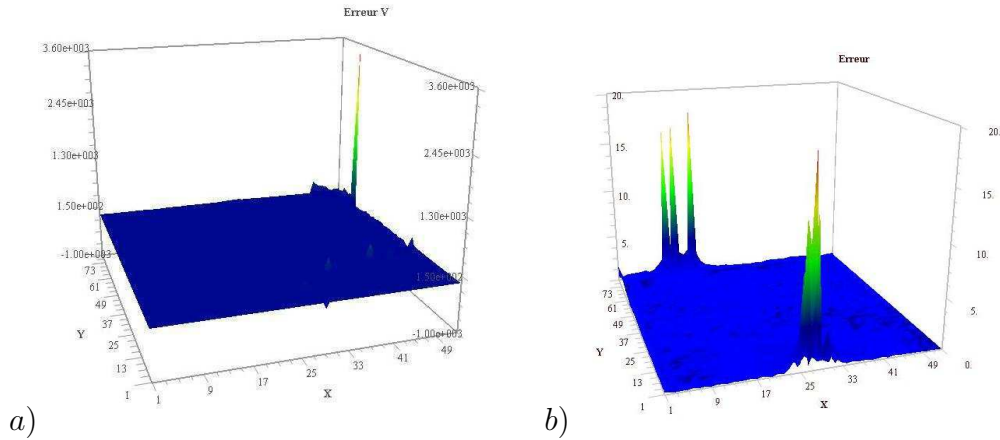
Nous allons maintenant voir si le relief influence cette estimation.



TAB. 5.3 – Illustration de l'estimation de la condition initiale pour la cas test d'un bassin a fond plat. a- La condition initiale donnée sur la vitesse V (en $\text{m}^3.\text{s}^{-1}$). Cette condition initiale est celle que l'assimilation de donnée variationnelle doit nous aider à retrouver. b- La condition initiale en V (en $\text{m}^3.\text{s}^{-1}$) permettant de démarrer l'exécution du code d'optimisation, celle-ci est assez éloignée de celle que nous devons retrouver. c- La condition initiale en V (en $\text{m}^3.\text{s}^{-1}$) obtenue grâce à l'aide de la minimisation de la fonction coût.



TAB. 5.4 – Illustration de l'estimation de la condition initiale pour la cas test d'un bassin a fond plat. a- La condition initiale donnée sur la hauteur d'eau Zn (en m). Cette condition initiale est celle que l'assimilation de donnée variationnelle doit nous aider à retrouver. b- La condition initiale sur la hauteur d'eau Zn (en m) permettant de démarrer l'exécution du code d'optimisation, celle-ci est assez éloignée de celle que nous devons retrouver. c- La condition initiale en Zn (en m) obtenue grâce à l'aide de la minimisation de la fonction coût.



TAB. 5.5 – Illustration de l'efficacité de l'estimation de la condition initiale pour la cas test d'un bassin a fond plat. a) Erreur relative commise lors de l'estimation de la condition initiale en V (en $\text{m}^3.\text{s}^{-1}$), dans le cas d'un domaine représentant un bassin à fond plat. Etude menée avec le logiciel LMCFLD. b) Erreur relative commise lors de l'estimation de la condition initiale en ZN (en m), dans le cas d'un domaine représentant un bassin à fond plat. Etude menée avec le logiciel LMCFLD.

Un bassin dont le fond est légèrement incliné

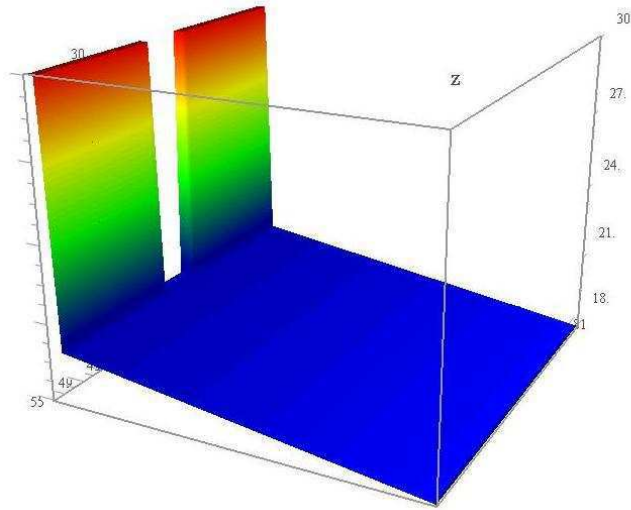
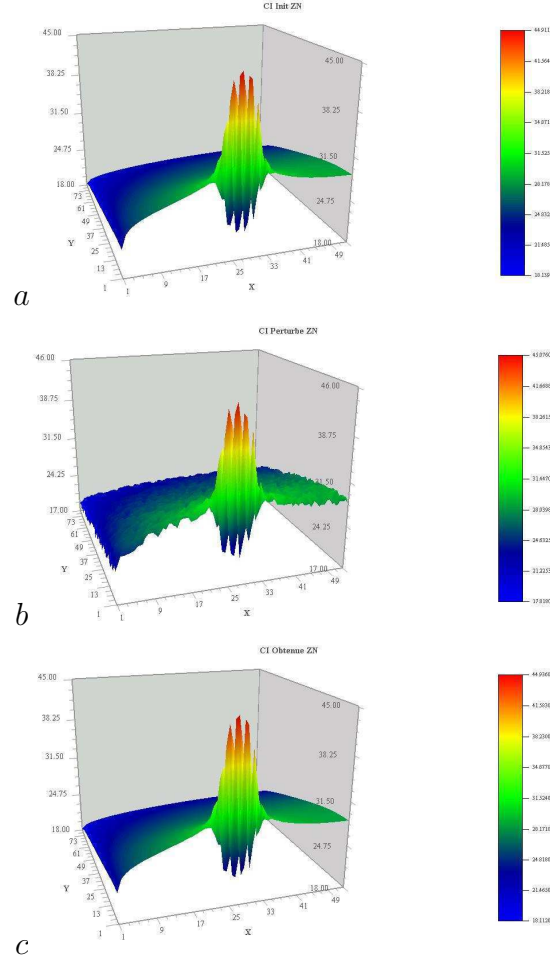


FIG. 5.5 – Représentation du bassin dont le fond est en pente légère.

Le domaine étudié ici est formé d'un bassin carré ($50 \text{ m} \times 50 \text{ m}$) dont la partie droite est bordée d'une digue infranchissable. Le fond de ce bassin est en pente légère à 2% (cf figure 5.5). L'eau se déverse par une brèche rectangulaire pratiquée dans la digue.

La condition aux limites ouest sera donc définie grâce à des couples (hauteur, temps).



TAB. 5.6 – Illustration de l'estimation de la condition initiale pour la cas test d'un bassin dont le fond est légèrement en pente. a- La condition initiale donnée sur la hauteur d'eau, Zn. Cette condition initiale est celle que l'assimilation de donnée variationnelle doit nous aider à retrouver. b- La condition initiale en Zn permettant de démarrer l'exécution du code d'optimisation, celle-ci est assez éloignée de celle que nous devons retrouver. c- La condition initiale en Zn obtenue grâce à l'aide de la minimisation de la fonction coût.

Les conditions aux limites nord et sud sont de type glissement sans frottement :

$$\vec{u} \cdot \vec{n} = 0$$

et la condition aux limites est de type frontière ouverte

$$\frac{\partial \vec{u}}{\partial \vec{n}} = 0$$

Le temps de l'écoulement sera de 1000 secondes avec un pas de temps de 2 secondes.

Afin de tester les méthodes d'assimilation de données, nous avons fait appel à la méthode des expériences jumelles pour obtenir un jeu d'observations sur les hauteurs d'eau et les vitesses.

Comme nous l'avons fait précédemment, il faut maintenant définir une fonction coût à

minimiser :

$$J(CI) = \frac{1}{2}\|U - U_{obs}\|^2 + \frac{1}{2}\|V - V_{obs}\|^2 + \frac{1}{2}\|Zn - Zn_{obs}\|^2 \quad (5.4)$$

où (U, V) sont les coordonnées de la vitesse, Zn la hauteur d'eau et X_{obs} les observations fournies par la méthode des expériences jumelles (avec $X = U, V, Zn$). Les observations sont disponibles au temps final.

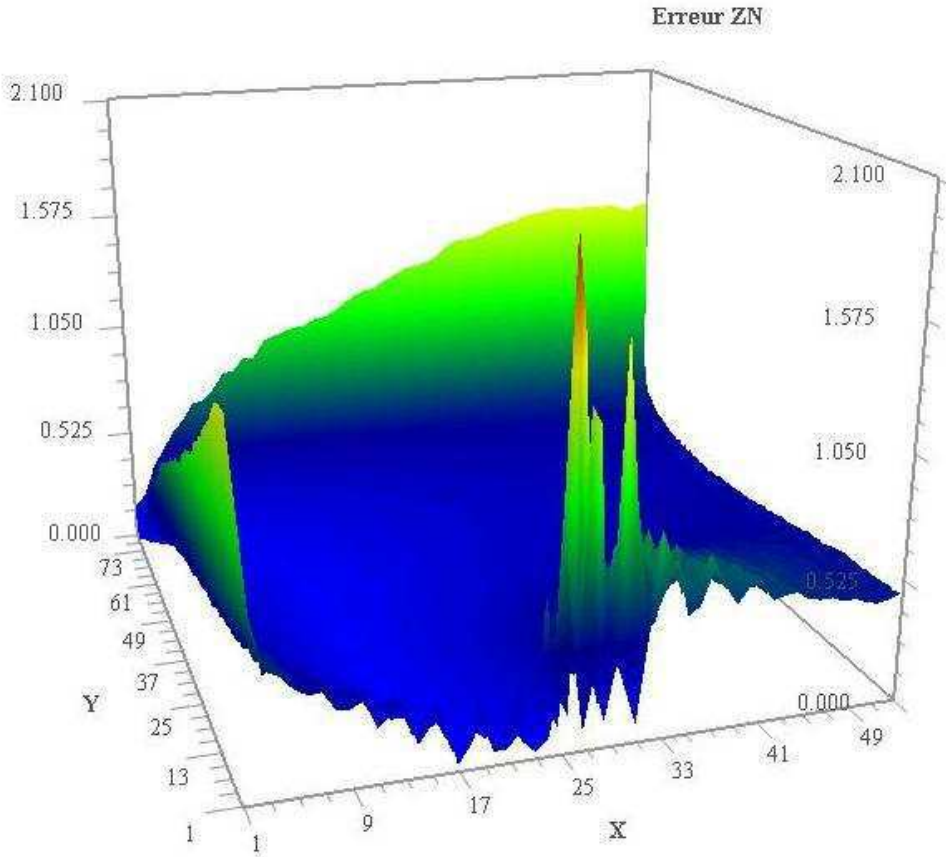


FIG. 5.6 – *Erreur commise lors de l'estimation de la condition initiale donnée en ZN sur le domaine représentant un bassin dont le fond est en pente. Étude menée avec le logiciel LMCFLD.*

La condition initiale, dans les mêmes conditions que précédemment, est tout aussi bien retrouvée (cf figure c dans le tableau 5.6). L'erreur (cf tableau 5.6) paraît toutefois plus importante dans cet exemple que dans le précédent en restant toutefois dans des limites raisonnables puisque l'erreur maximale commise est de 2.0%.

Dans toute la suite de ce travail, nous étudierons les différentes utilisation du code adjoint grâce à celui de LMCFLD.

5.2 Estimation des conditions aux limites

5.2.1 Trois exemples de condition aux limites

Soit un bassin carré (50 m × 50 m) dont la face ouest est bordée par une digue insubmersible. Le fond de ce bassin est plat et la nature du sol est homogène. Les conditions aux limites nord et sud sont de type glissement sans frottement :

$$\vec{U} \cdot \vec{n} = 0$$

et la condition aux limites est de type frontière ouverte

$$\frac{\partial \vec{U}}{\partial \vec{n}} = 0$$

Dans la digue à l'ouest, nous pratiquons une brèche par laquelle l'eau va s'infiltrer pour alimenter le bassin. C'est cette condition aux limites que nous allons estimer, condition définie par une succession de couples (hauteur d'eau, temps).

Le temps de l'écoulement sera de 1000 secondes avec un pas de temps de 2 secondes.

Sur ce domaine prédéfini, nous allons utiliser trois formes de conditions aux limites grâce auxquelles l'étude de la pertinence de l'assimilation de données variationnelle sera menée. Pour cela, nous devons définir une fonction coût à minimiser :

$$J(CL) = \frac{1}{2} \|U - U_{obs}\|^2 + \frac{1}{2} \|V - V_{obs}\|^2 + \frac{1}{2} \|Zn - Zn_{obs}\|^2 \quad (5.5)$$

où (U, V) sont les coordonnées de la vitesse, Zn la hauteur d'eau et X_{obs} les observations fournies par la méthode des expériences jumelles (avec $X = \{U, V, Zn\}$). Les observations sont disponibles à chaque pas de temps et pour chaque point de la grille.

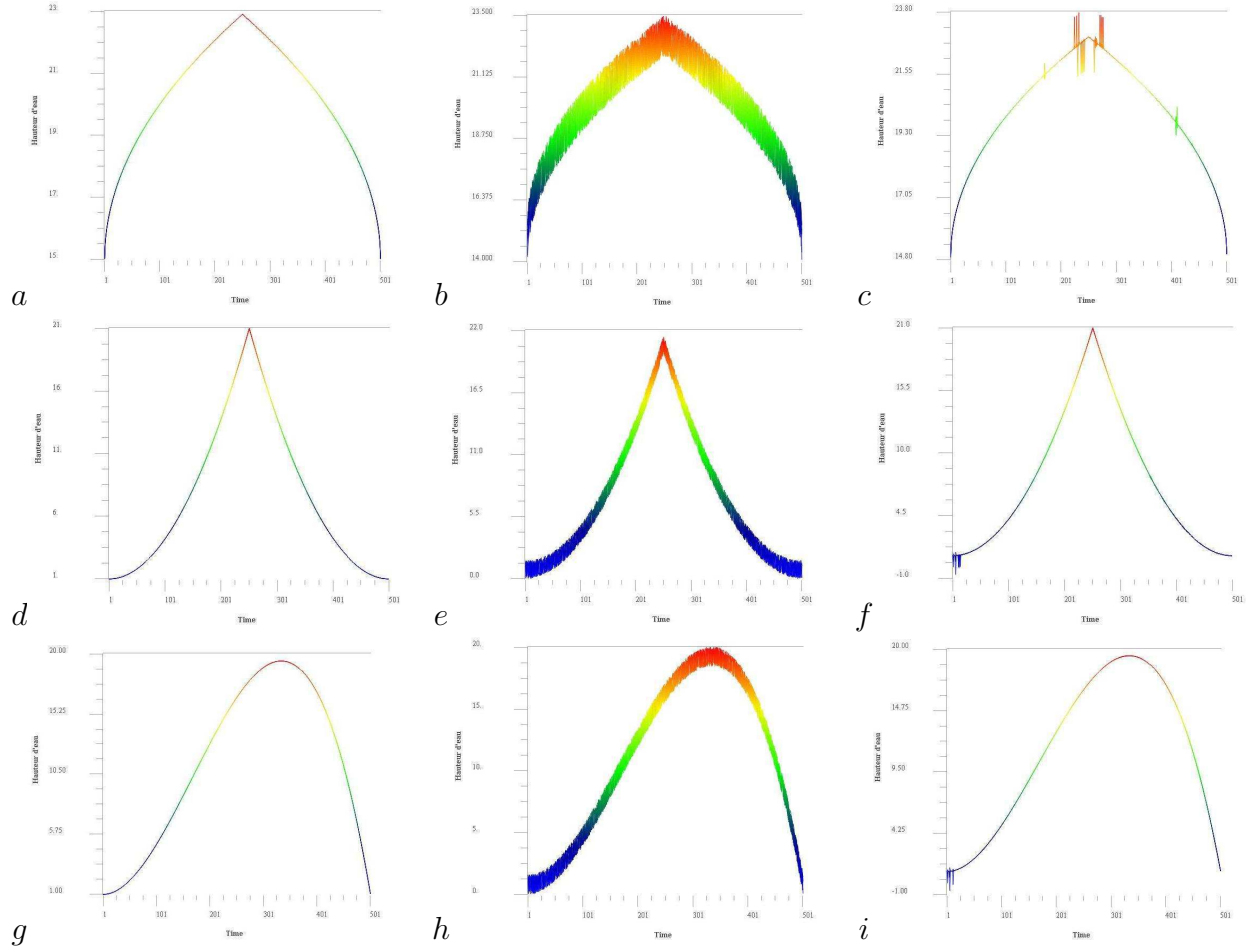
Sur un domaine dont le relief est uniforme, le tableau 5.7 montre la qualité de l'estimation de la condition aux limites grâce à la minimisation de la fonction coût pour un jeu d'observation dense et ce pour les trois exemples de condition aux limites. En effet, quel que soit la vitesse à laquelle on injecte de l'eau dans le bassin, la précision de la condition retrouvée est plus que satisfaisante puisque l'erreur est inférieure à 10^{-2} dans la plupart des cas (voir le tableau 5.8).

Nous avons travaillé avec un bassin dont le fond est plat. On peut maintenant se demander si le relief du fond du bassin peut avoir une influence sur l'estimation de la condition aux limites. Effectivement, plus la pente du fond sera forte plus le flux sera proche d'un passage en torrentiel. Cette accélération chaotique de l'eau peut-elle générer une erreur de l'estimation ?

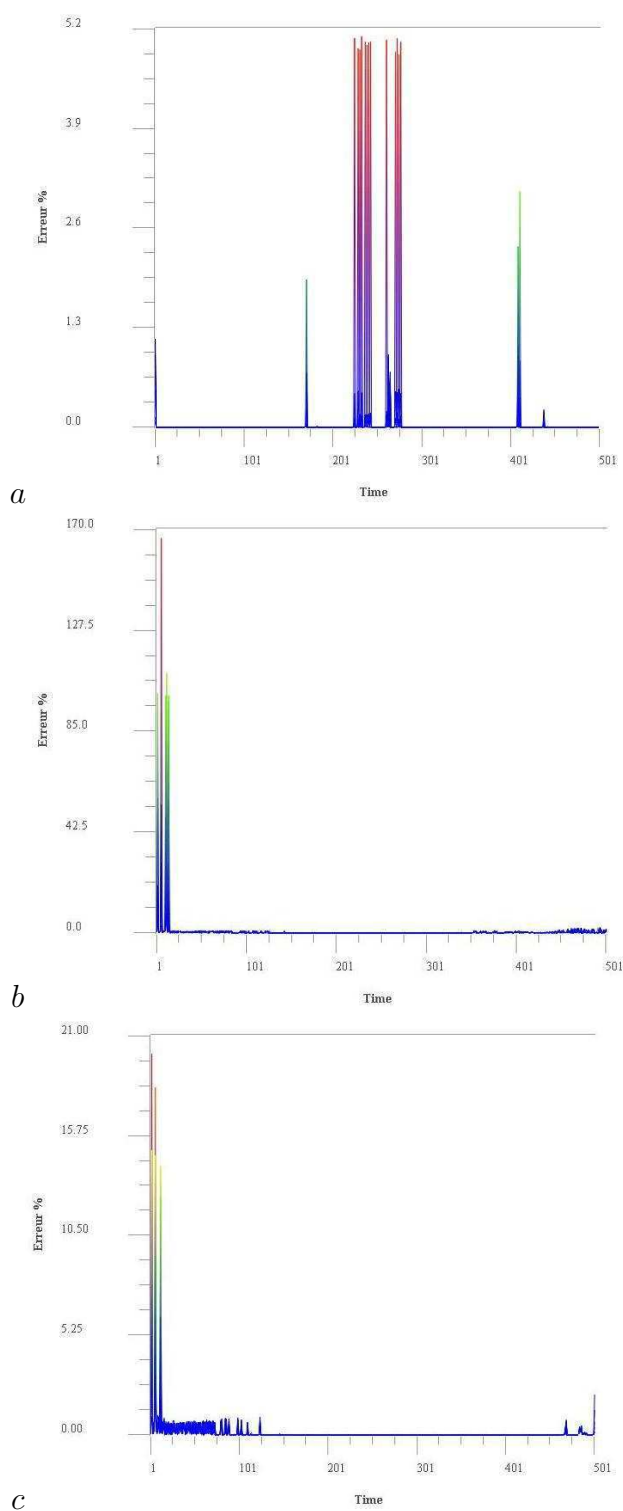
5.2.2 Influence du relief dans l'estimation de la condition aux limites

Soit une rivière linéaire (50 m) bordée par deux digues insubmersibles. Le coefficient de rugosité est le même sur toute la zone. La condition aux limites aval est de type frontière ouverte

$$\frac{\partial \vec{u}}{\partial \vec{n}} = 0$$



TAB. 5.7 – Illustration de l'estimation de la condition aux limites pour la cas test d'un bassin dont le fond est légèrement en pente. *a*, *d* et *g*- Les conditions aux limites à retrouver grâce à la méthode d'assimilation de données dédiée à l'estimation de la condition aux limites développée précédemment. *b*, *e* et *i*- Les conditions aux limites données pour initialiser l'exécution de l'adjoint. *c*, *f* et *i*- Les Conditions aux limites obtenues grâce à la minimisation de la fonction coût.



TAB. 5.8 – *Illustration de l'erreur de l'estimation de la condition aux limites dans un bassin plat. Ces courbes représentent les erreurs commise par la méthode de l'adjoint afin de retrouver les conditions aux limites optimales dans les trois cas exposés précédemment.*

C'est la condition aux limites amont que nous allons estimer. Elle est définie par une succession de couples (hauteur d'eau, temps).

Le temps de l'écoulement sera de 1000 secondes avec un pas de temps de 2 secondes.

Sur ce domaine prédéfini, nous allons utiliser deux formes de conditions aux limites grâce auxquelles on étudiera la fiabilité de l'estimation de ce paramètre lorsque la pente du lit de la rivière devient de plus en plus importante.

Pour cela, nous devons définir une fonction coût à minimiser :

$$J(CL) = \frac{1}{2} \|Zn - Zn_{obs}\|^2 \quad (5.6)$$

où Zn la hauteur d'eau et Zn_{obs} les observations fournies par la méthode des expériences jumelles. Les observations sont disponibles à chaque pas de temps et pour chaque point de la grille.

Le tableau 5.9 montre l'erreur commise lors de l'estimation de la condition aux limites pour trois types de relief : un lit sans relief, un autre avec une pente assez petite de l'ordre de 4% et le dernier avec une pente plus forte, environ 16%.

Au premier abord, il semblerait que l'erreur commise est inversement proportionnelle à l'accroissement de la pente. En effet, nous pouvons voir dans les figures du tableau précédent que la densité du nombre de pics d'erreur décroît avec l'augmentation de la pente. Mais la différence étant infime, on peut se demander si c'est réellement le fait de l'influence de la pente sur l'estimation de la condition aux limites ou bien une erreur d'arrondi ou peut être encore le fait que nous sommes à la limite des compétences des équations de Saint Venant (flux torrentiel).

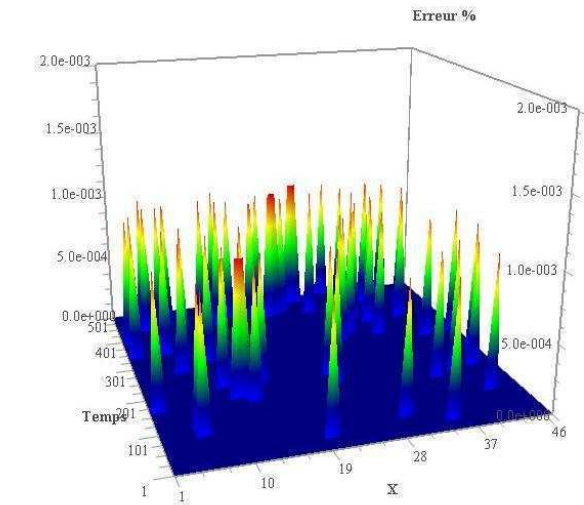
Contrairement au premier exemple de cette section, la condition aux limites utilisée ici n'est pas constante dans le temps. Effectivement, nous reprenons ici la condition aux limites introduite dans la section précédente (exemple a). Les figures du tableau 5.10 nous montrent clairement que quel que soit la pente du fond de la rivière, l'erreur commise en retrouvant la condition aux limites optimale est équivalente. *A priori*, la pente du fond de la rivière ne paraît n'avoir aucune influence sur la qualité de l'estimation de la condition aux limites.

Dans les exemples traités antérieurement, nous avons utilisé une fonction coût comparant les résultats du modèle direct avec un jeu de données dense et complet, c'est-à-dire utilisant les observations en V , U et Zn . La fonction coût (5.6) utilisée pour cette étude, requiert des observations, certes denses, mais surtout limitées à la hauteur d'eau Zn . Nous pouvons voir que ce type de dégradation des observations ne dégrade pas les résultats de l'estimation de la condition aux limites. Il nous reste à étudier la dégradation de la densité des données, ce que nous ferons ultérieurement.

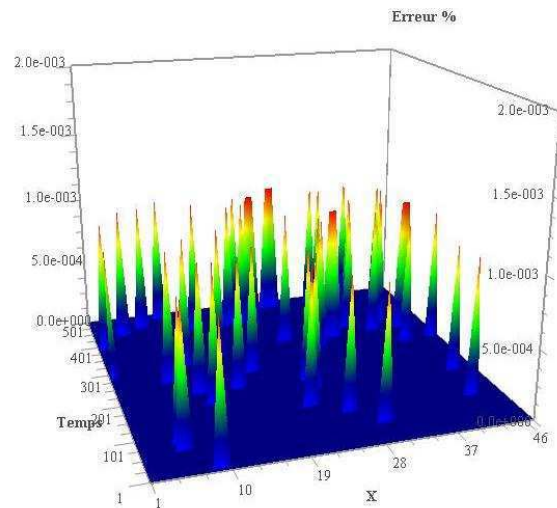
5.3 Couplage CI et CL

Dans cet exemple nous allons faire l'estimation conjointe de la condition initiale et de la condition aux limites imposée dans la brèche.

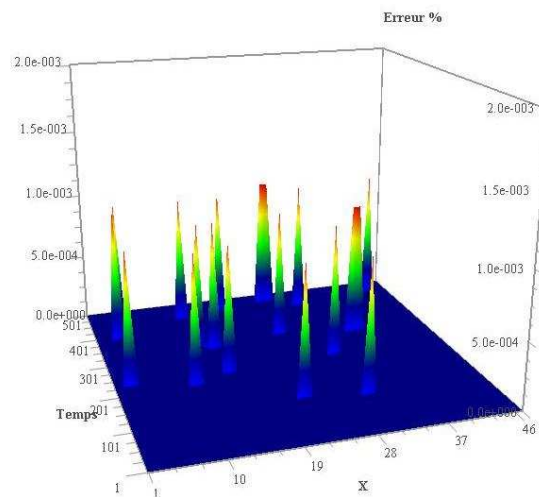
Soit un bassin carré (50 m × 50 m) dont la face ouest est bordée par une digue insubmersible. Le fond de ce bassin est plat et la nature du sol est homogène. Les conditions aux



a

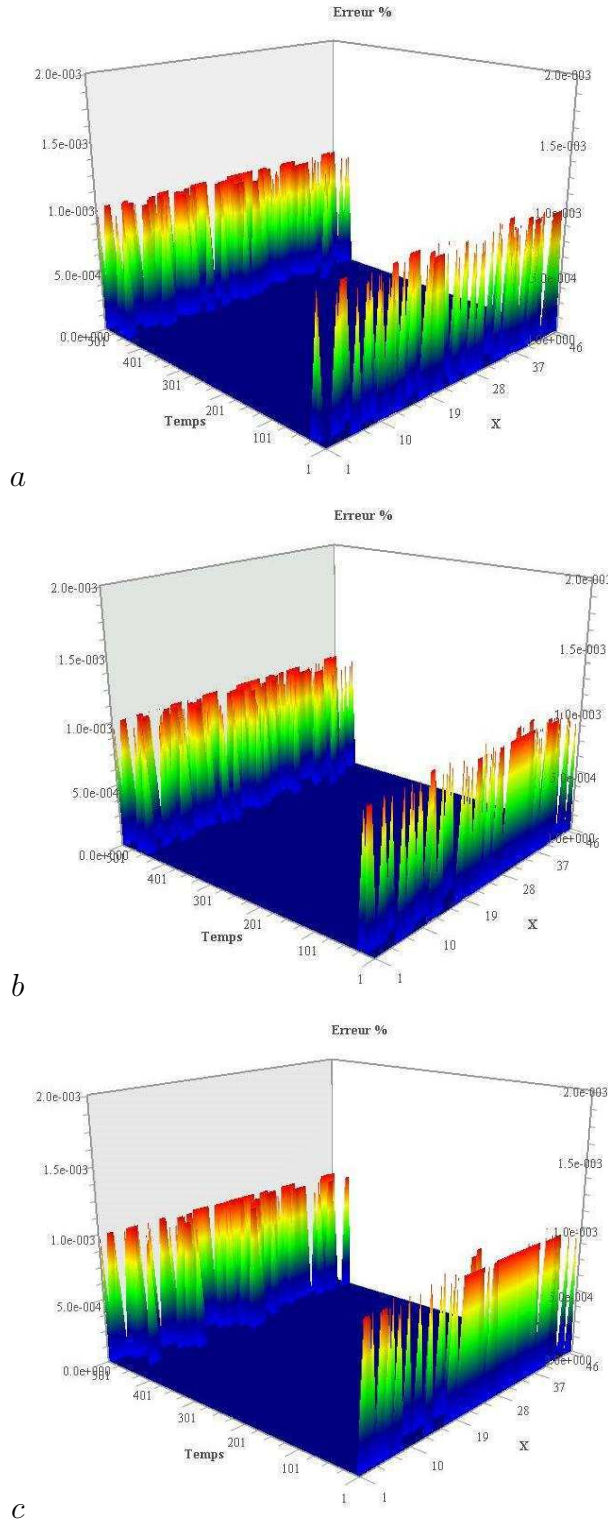


b



c

TAB. 5.9 – Illustration de l'erreur de l'estimation de la condition aux limites dans un bassin plat. Ces courbes représentent les erreurs commises par la méthode de l'adjoint afin de retrouver la condition aux limites optimale suivant la pente du fond de la rivière. La condition aux limites étudiée dans ces cas est constante dans le temps.



TAB. 5.10 – Illustration de l'erreur de l'estimation de la condition aux limites dans un bassin plat. Ces courbes représentent les erreurs commises par la méthode de l'adjoint afin de retrouver la condition aux limites optimale suivant la pente du fond de la rivière. La condition aux limites étudiée dans ces cas est celle étudiée dans la section précédente sous la dénomination a.

limites nord et sud sont de type glissement sans frottement :

$$\vec{u} \cdot \vec{n} = 0$$

et la condition aux limites est de type frontière ouverte

$$\frac{\partial \vec{u}}{\partial \vec{n}} = 0$$

Dans la digue à l'ouest, nous pratiquons une brèche par laquelle l'eau va s'infiltrer pour alimenter le bassin. Cette condition aux limites est définie par une succession de couples (hauteur d'eau, temps).

Le temps de l'écoulement sera de 1000 secondes avec un pas de temps de 2 secondes.

Au temps initial, le fond de cette digue est sec.

Nous devons définir une fonction coût à minimiser :

$$J(CI, CL) = \frac{1}{2} \|U - U_{obs}\|^2 + \frac{1}{2} \|V - V_{obs}\|^2 + \frac{1}{2} \|Zn - Zn_{obs}\|^2 + \frac{1}{2} \|CI - CI_0\|^2 + \frac{1}{2} \|CL - CL_0\|^2 \quad (5.7)$$

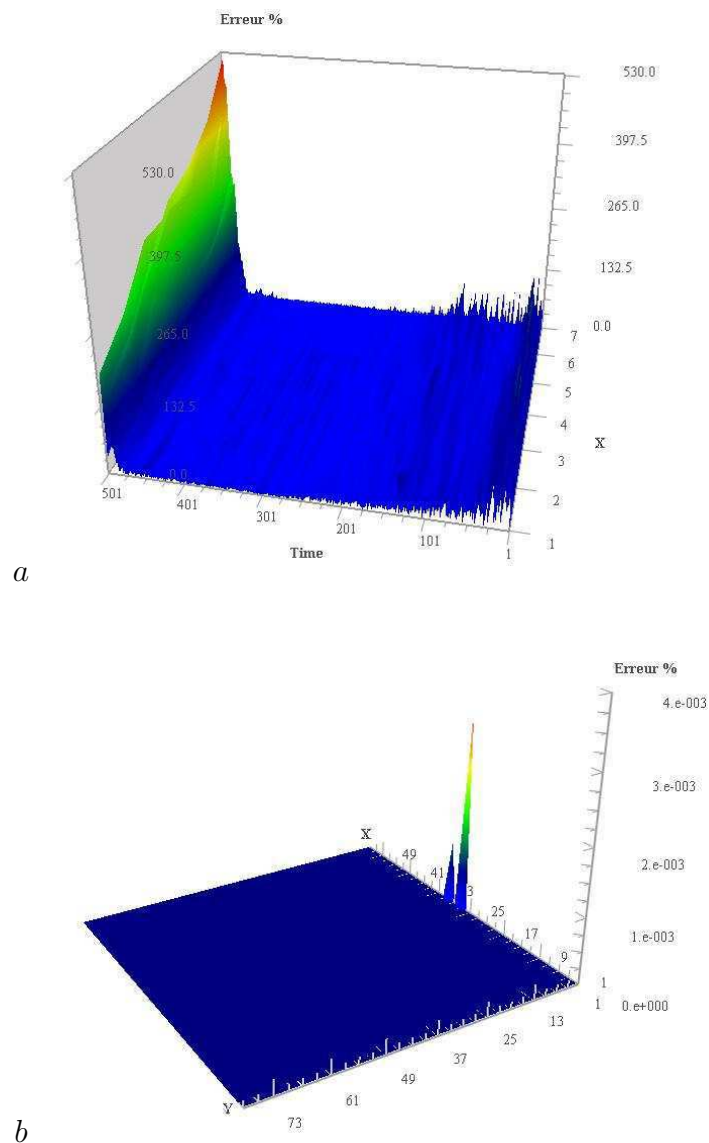
où (U, V) sont les coordonnées de la vitesse, Zn la hauteur d'eau et X_{obs} les observations fournies par la méthode des expériences jumelles (avec $X = U, V, Zn$). Les observations sont disponibles à chaque pas de temps et pour chaque point de la grille.

De plus, nous avons ajouté deux termes de forçages afin d'orienter la minimisation de J vers son minimum global. Ainsi le premier terme calcule la différence entre la condition initiale obtenue et la condition initiale attendue. Le terme CI_0 est ici la condition initiale à retrouver et non pas une idée *a priori* de sa valeur comme il est classique de faire dans une approche opérationnelle. Le deuxième terme mesure la différence entre la condition aux limites calculée et la condition aux limites attendue. CL_0 est ici la condition aux limites à retrouver.

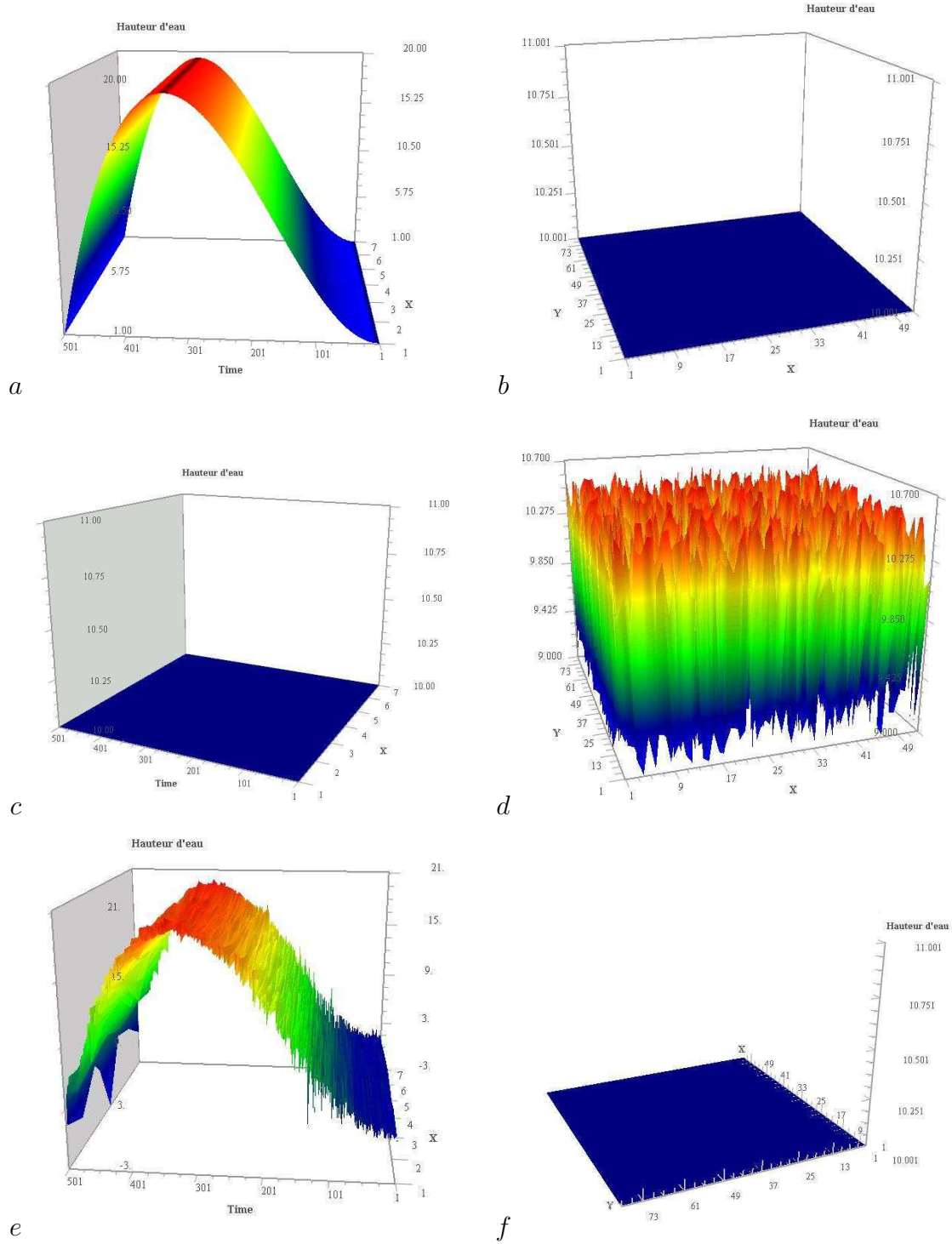
Le but de cet exemple sera de montrer dans quelles conditions se fait l'estimation conjointe de la condition initiale et de la condition aux limites.

Estimer en même temps deux paramètres d'entrée du modèle implique forcément que les résultats obtenus vont contenir une part d'erreur supplémentaire. Ainsi, dans cet exemple, l'estimation de la condition initiale fournit une solution très proche de celle attendue (cf figure f dans le tableau 5.12). Toutefois, contrairement aux exemples présentés précédemment, l'estimation de la condition aux limites est assez éloignée de la perfection (cf figure e dans le tableau 5.12).

En effet, à partir d'une condition aux limites très éloignée de celle à retrouver (voir figure c du tableau 5.12), nous ne sommes capables que de retrouver la forme de la courbe imposée en entrée, mais plus les valeurs exactes composant cette courbe (voir figure e dans le tableau 5.12). Toutefois l'erreur commise est assez petite dans l'ensemble pour être satisfaisante (cf figure a et b dans le tableau 5.11). Toutefois, l'estimation du pic de la crue est très précis. Cette connaissance étant souvent la plus recherchée dans une étude hydraulique, le résultat fourni par la minimisation de la fonction coût est prometteur.



TAB. 5.11 – Illustration de l'erreur de l'estimation conjointe de la condition aux limites et de la condition initiale dans un bassin plat. Ces courbes représentent les erreurs commises par la méthode de l'adjoint afin de retrouver les conditions aux limites (figure a) et initiale (figure b) optimales dans les trois cas exposés précédemment.



TAB. 5.12 – Illustration de l'estimation conjointe de la condition aux limites et de la condition initiale dans un bassin plat. a (resp b-) La courbe représente la condition aux limites (resp. initiale) à retrouver grâce à la méthode d'assimilation de données dédiée à l'estimation de la condition aux limites et de la condition initiale développée précédemment. c (resp. d-) La condition aux limites (resp. initiale) donnée pour initialiser l'exécution de l'adjoint. e (resp. f-) La Condition aux limites (resp. initiale) obtenue grâce à la minimisation de la fonction coût.

5.4 Conclusion

Dans cette partie, nous avons vu que l'utilisation du système adjoint pour minimiser une fonction coût mesurant l'écart entre les observations et les résultats du code direct nous donne des résultats très prometteurs.

L'estimation de la condition initiale est un point crucial dans la domaine de la prévision. Elle permet de reconstituer de manière optimale l'état présent, afin de repartir sur des bases saines. Dans les quelques cas présentés, nous avons montré que l'estimation de la condition initiale donne des résultats très intéressants qui combinent performance et précision. Toutefois reste le problème de la densité des observations qui semble peu réaliste. En effet, en hydraulique la plupart des observations sont faites sur les hauteurs d'eau et non pas sur les vitesses; de plus avoir une observation en chaque point de la grille ne paraît pas raisonnable. Ce cas précis de dégradation des observations sera traité dans une section ultérieurement.

L'estimation de cette condition aux limites est un point crucial dans l'étude des crues. Il est toujours primordial de connaître la quantité d'eau injectée dans un bassin versant qui est à l'origine de telles ou telles inondations. A la suite d'une catastrophe pareille, il reste toujours des marques sur les maisons ou sur les arbres. A partir de ces observations, il semble raisonnable de penser que nous pourrions reconstituer la condition aux limites ou toutefois retrouver la valeur du pic de la crue car l'impossibilité de dater ces marques laissées par l'eau est un problème qui ne nous permet pas de reconstituer entièrement la condition aux limites.

La modélisation d'une rivière nécessite la connaissance de la condition initiale ET de la condition aux limites. Celles ci étant souvent méconnues, il est nécessaire d'en faire l'estimation. Avec un jeu d'observations ces estimations ne peuvent se mener que de façon conjointe. Le dernier exemple présenté nous montre un premier résultat intéressant. Toutefois, c'est l'estimation de la condition aux limites qui semble poser le plus de problèmes et qui doit donc être au centre des préoccupations futures.

Chapitre 6

Assimilation de données et identification de paramètres

Dans ce chapitre, nous allons voir l'application du modèle adjoint pour l'identification des paramètres d'une modélisation. Le premier de ces paramètres est le coefficient de rugosité dont la formule empirique exprime la dissipation distribuée d'énergie.

Cette expression cherche à calculer de manière schématique le frottement du fluide sur le fond, mais aussi un ensemble de phénomènes de dissipation d'énergie dont l'échelle spatiale est inférieure à la dimension caractéristique du maillage de calcul.

La difficulté d'appréciation de ce coefficient de rugosité provient de son caractère globalisateur, nous allons tenter de proposer par l'assimilation de données une alternative à la détermination experte de sa valeur.

Le deuxième de ces paramètres est la géométrie même. Il s'agit ici de tenter de répondre à plusieurs questions pratiques. On rappelle certaines de ces questions posées en introduction :

- la schématisation introduite par la discrétisation ;
- le caractère discret des mesures topographiques ;
- l'incertitude sur la connaissance de la bathymétrie des parties inondées.

Les données utilisées pour la construction des exemples proviennent de deux sources distinctes :

- virtuelles, afin d'établir des cas tests intéressants qui permettent de se focaliser sur des particularités physiques dont l'influence sur l'écoulement est significatif.
- réelles, afin de montrer les possibilités des méthodes d'assimilation de données dans des cas où de nombreux problèmes sont mélangés.

Cependant, quelle que soit l'origine des données utilisées pour la construction des modèles, aucun exemple présenté n'utilise d'observations réelles. En effet, comme il a déjà été dit dans les chapitres précédents, le manque d'observations en hydraulique est trop important pour tester les méthodes d'assimilation de données de manière optimale.

Les démonstrations qui suivent sont réalisées avec le modèle bidimensionnel LMCFLD. Dans ce modèle, le rugosité est représenté par le coefficient de Manning. Dans la deuxième partie de ce chapitre, nous étudierons quelques exemples sur l'estimation de la topographie équivalente.

6.1 Estimation de la rugosité

6.1.1 Un simple bassin

Pour débiter cette étude sur l'estimation de la rugosité, nous allons mettre en place un exemple avec le logiciel LMCFLD.

Soit un domaine carré ($50m \times 50m$) bordé à l'ouest par une digue¹. Le fond de ce bassin est plat. Au milieu de la digue se trouve une brèche par laquelle l'eau s'infiltré dans le bassin. La condition aux limites ouest sera donc définie grâce à des couples (hauteur, temps).

Les conditions aux limites nord et sud sont de type glissement sans frottement :

$$\vec{u} \cdot \vec{n} = 0$$

et la condition aux limites est de type frontière ouverte

$$\frac{\partial \vec{u}}{\partial \vec{n}} = 0$$

Le temps de l'écoulement sera de 1000 secondes avec un pas de temps de 2 secondes.

Le jeu de Manning, représentant la nature du sol, est uniformément égal à 0.05 (cf figure a dans le tableau 6.1). Le but de cette étude sera de montrer que l'estimation de ce paramètre est possible grâce à un jeu d'observations sur la zone, fourni par la méthode des expériences jumelles. La fonction coût à minimiser dans cet exemple est

$$J(Manning) = \frac{1}{2} \|U - U_{obs}\|^2 + \frac{1}{2} \|V - V_{obs}\|^2 + \frac{1}{2} \|Zn - Zn_{obs}\|^2 \quad (6.1)$$

où (U, V) sont les coordonnées de la vitesse, Zn la hauteur d'eau et X_{obs} les observations fournies par la méthode des expériences jumelles (avec $X = \{U, V, Zn\}$). Les observations sont disponibles au temps $t = 1000$.

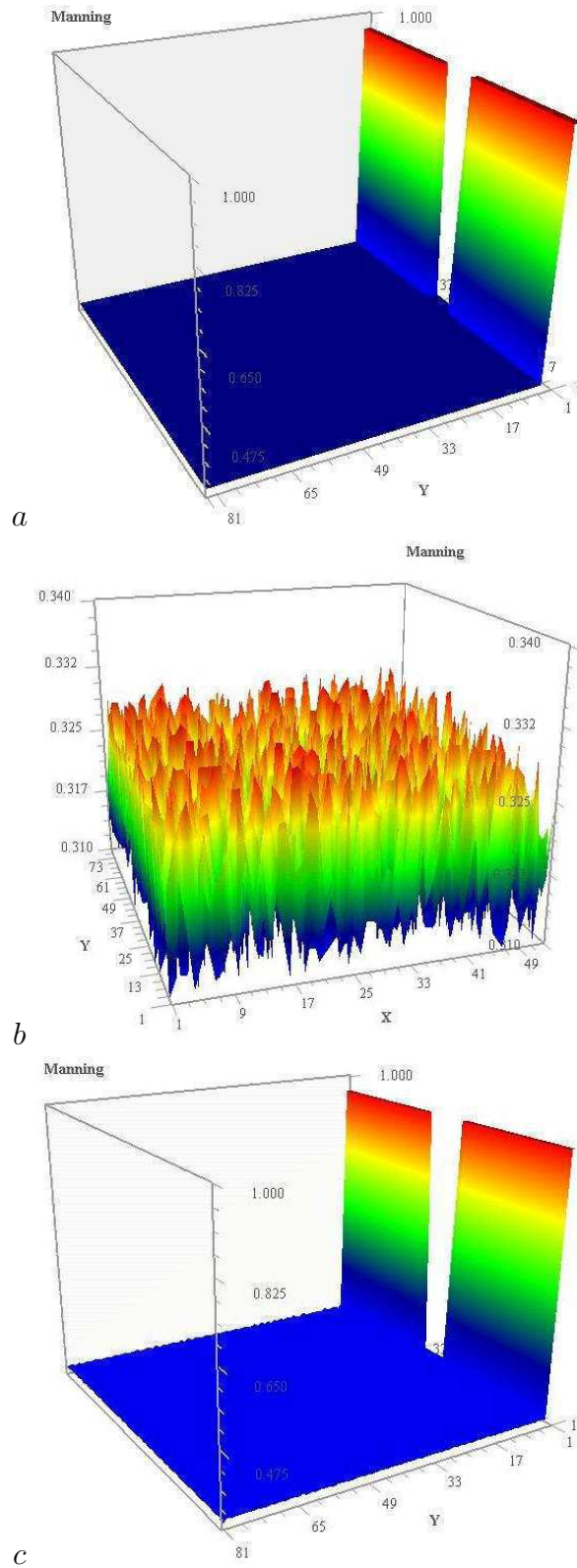
Afin d'initialiser l'exécution du code d'optimisation, on perturbe de façon aléatoire le jeu de Manning uniforme pour trouver un nouveau jeu très hétérogène (cf figure b dans le tableau 6.1). Après 15 itérations du code d'optimisation, la fonction coût tend vers 0 (cf la figure b dans la tableau 6.2), un nouveau jeu de Manning est alors retrouvé.

Ce jeu de Manning est très proche de celui que nous cherchions à identifier. En effet, l'erreur maximale commise est de l'ordre de 2% (voir figure a dans la tableau 6.2).

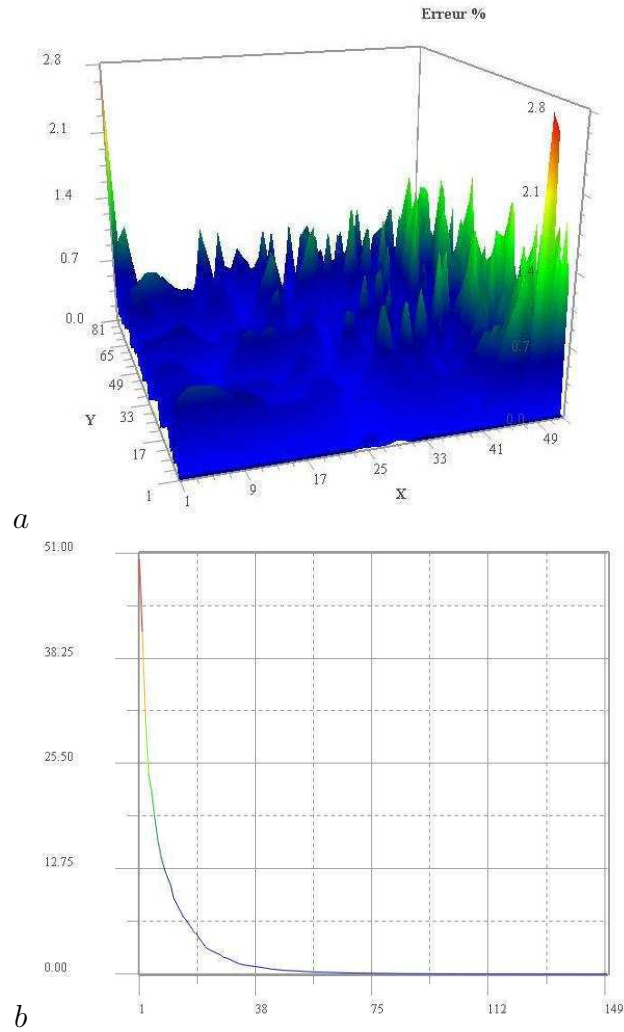
Cet exemple, nous permet d'illustrer de manière simple la qualité avec laquelle un jeu de rugosité peut être trouvé à partir d'observations faites sur les vitesses et les hauteurs d'eau.

Nous allons maintenant étudier le même type d'estimation mais pour un jeu de Manning légèrement plus complexe.

¹Dans ce mémoire de thèse nous essayerons de garder la même orientation pour représenter graphiquement le bassin. A savoir que la frontière ouest, bordée par la digue, sera représentée sur le côté droit du schéma. Cette orientation a été choisie pour des raisons de clarté des figures. De même, les représentations graphique du jeu de Manning seront toujours orientées de la même façon. Lorsqu'elles sont vues de dessus, la digue se situe en bas de la figure



TAB. 6.1 – Illustration de l'estimation de la rugosité dans un bassin plat. Le coefficient de rugosité sur la digue a été choisi très grand de manière à la faire ressortir dans la représentation graphique. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est celle que l'assimilation de données variationnelle doit nous aider à retrouver. b- La rugosité permettant de démarrer l'exécution du code d'optimisation, celle-ci est assez éloignée de celle que nous devons retrouver. c- La rugosité obtenue grâce à la minimisation de la fonction coût.



TAB. 6.2 – *Illustration de l'efficacité de l'estimation de la rugosité dans un bassin plat. a- La différence entre le jeu de rugosité initial et celui obtenu grâce à la minimisation de la fonction coût.. b- Evolution de la valeur de la fonction coût (6.1) en fonction des itérations du code d'optimisation.*

6.1.2 Un jeu de Manning binaire

Soit un domaine carré ($50m \times 50 m$) bordé à l'ouest par une digue. Le fond de ce bassin est plat. Au milieu de la digue se trouve une brèche par laquelle l'eau s'infiltre dans le bassin. La condition aux limites ouest sera donc définie grâce à des couples (hauteur, temps).

Les conditions aux limites nord et sud sont de type glissement sans frottement :

$$\vec{u} \cdot \vec{n} = 0$$

et la condition aux limites est de type frontière ouverte

$$\frac{\partial \vec{u}}{\partial \vec{n}} = 0$$

Le temps de l'écoulement sera de 1000 avec un pas de temps de 2.

Le jeu de Manning, représentant la nature du sol, est uniformément égal à 0.05 excepté sur une petite zone au centre du bassin.

La fonction coût utilisée dans cet exemple est

$$J(Manning) = \frac{1}{2} \|U - U_{obs}\|^2 + \frac{1}{2} \|V - V_{obs}\|^2 + \frac{1}{2} \|Zn - Zn_{obs}\|^2 \quad (6.2)$$

où (U, V) sont les coordonnées de la vitesse, Zn la hauteur d'eau et X_{obs} les observations fournies par la méthode des expériences jumelles (avec $X = \{U, V, Zn\}$). Les observations sont disponibles au temps $t = 1000$.

Cette fois ci une perturbation identique sur toute la zone (+0.01) va nous permettre d'initialiser l'exécution du code d'optimisation (cf figure b dans la tableau 6.3).

La figure b dans le tableau 6.4, nous montre l'évolution de la fonction coût tout au long de l'optimisation. La convergence de celle-ci vers 0 nous confirme la qualité du jeu de Manning obtenu, puisque le comportement du flux pour ce jeu de données est pratiquement identique à celui observé.

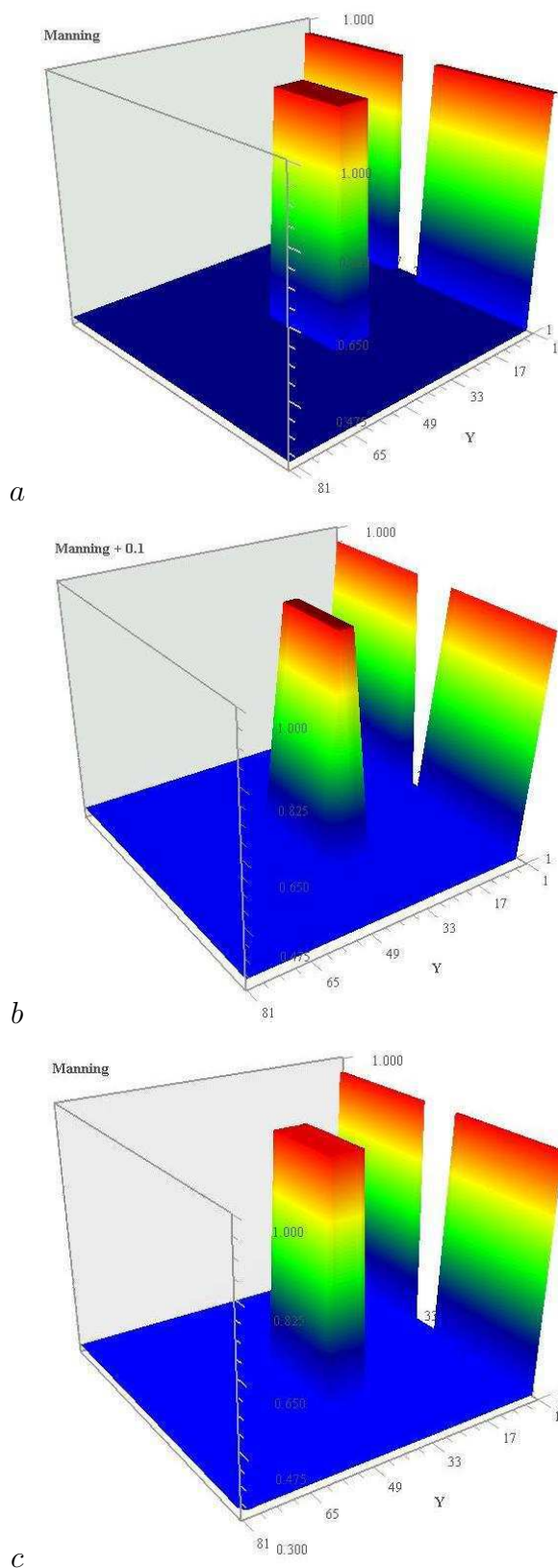
La figure a dans le tableau 6.4, illustrant l'erreur commise, montre clairement l'excellente précision obtenue (l'erreur est inférieure à 2%).

L'extension de la méthode de l'estimation du paramètre de rugosité à un jeu de Manning hétérogène, nous montre que la qualité démontrée dans la section précédente est conservée.

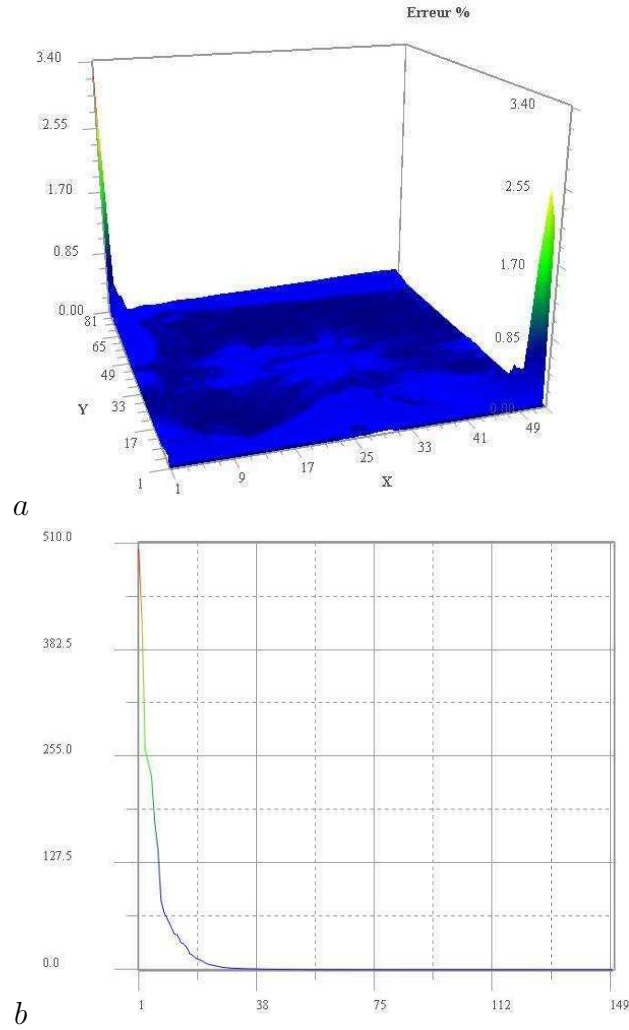
Mais le coefficient de Manning n'est pas un des coefficients qui influence le plus le comportement du flux. En effet, ce coefficient est de bien moindre importance par rapport à la topographie. Ainsi le relief de la zone peut-il dégrader l'estimation du jeu de Manning ?

6.1.3 Influence du relief

Soit un domaine carré ($50m \times 50 m$) bordé à l'ouest par une digue. Au milieu de la digue se trouve une brèche par laquelle l'eau s'infiltre dans le bassin. La condition aux



TAB. 6.3 – Illustration de l'estimation de la rugosité pour un jeu de Manning binaire. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est celle que l'assimilation de données variationnelle doit nous aider à retrouver. b- La rugosité permettant de démarrer l'exécution du code d'optimisation, celle-ci est assez éloignée de celle que nous devons retrouver. c- La rugosité obtenue grâce à la minimisation de la fonction coût.



TAB. 6.4 – *Illustration de l'efficacité de l'estimation de la rugosité pour un jeu de Manning binaire. a- La différence entre le jeu de rugosité initial et celui obtenu grâce à la minimisation de la fonction coût.. b- Evolution de la valeur de la fonction coût (6.1) en fonction des itérations du code d'optimisation.*

limites ouest sera donc définie grâce à des couples (hauteur, temps).

Les conditions aux limites nord et sud sont de type glissement sans frottement :

$$\vec{u} \cdot \vec{n} = 0$$

et la condition aux limites est de type frontière ouverte

$$\frac{\partial \vec{u}}{\partial \vec{n}} = 0$$

Le temps de l'écoulement sera de 1000 secondes avec un pas de temps de 2 secondes.

Le jeu de Manning, représentant la nature du sol, est uniformément égal à 0.05 excepté sur une petite zone au centre du bassin. Le fond du bassin est uniformément plat sauf à l'endroit où le Manning change, en ce lieu la topographie révèle la présence d'une hauteur. Cet exemple pourrait simuler la présence d'une colline boisée au milieu d'une zone de moindre rugosité.

La fonction coût utilisée dans cet exemple est

$$J(Manning) = \frac{1}{2} \|U - U_{obs}\|^2 + \frac{1}{2} \|V - V_{obs}\|^2 + \frac{1}{2} \|Zn - Zn_{obs}\|^2 \quad (6.3)$$

où (U, V) sont les coordonnées de la vitesse, Zn la hauteur d'eau et X_{obs} les observations fournies par la méthode des expériences jumelles (avec $X = \{U, V, Zn\}$). Les observations sont disponibles au temps $t = 1000$.

La perturbation pratiquée afin d'initialiser l'exécution du code d'optimisation est identique à celle utilisée précédemment (+0.01) (cf le figure b dans le tableau 6.5). La convergence de la fonction coût, visible dans la figure b du tableau 6.6, montre que, malgré le relief, l'estimation de la rugosité a été effectuée avec succès. De plus, l'erreur commise, illustrée dans la figure a du tableau 6.6, est pratiquement identique à celle obtenue précédemment. Ainsi, le relief ne semble pas mettre en cause la précision de l'estimation du coefficient de Manning.

Par conséquent, quelle que soit la nature du domaine étudié, l'estimation de Manning par l'utilisation du code adjoint peut être menée de manière performante. Et les résultats obtenus précédemment montrent que la qualité de l'estimation est excellente.

6.1.4 Schématisation du jeu de Manning

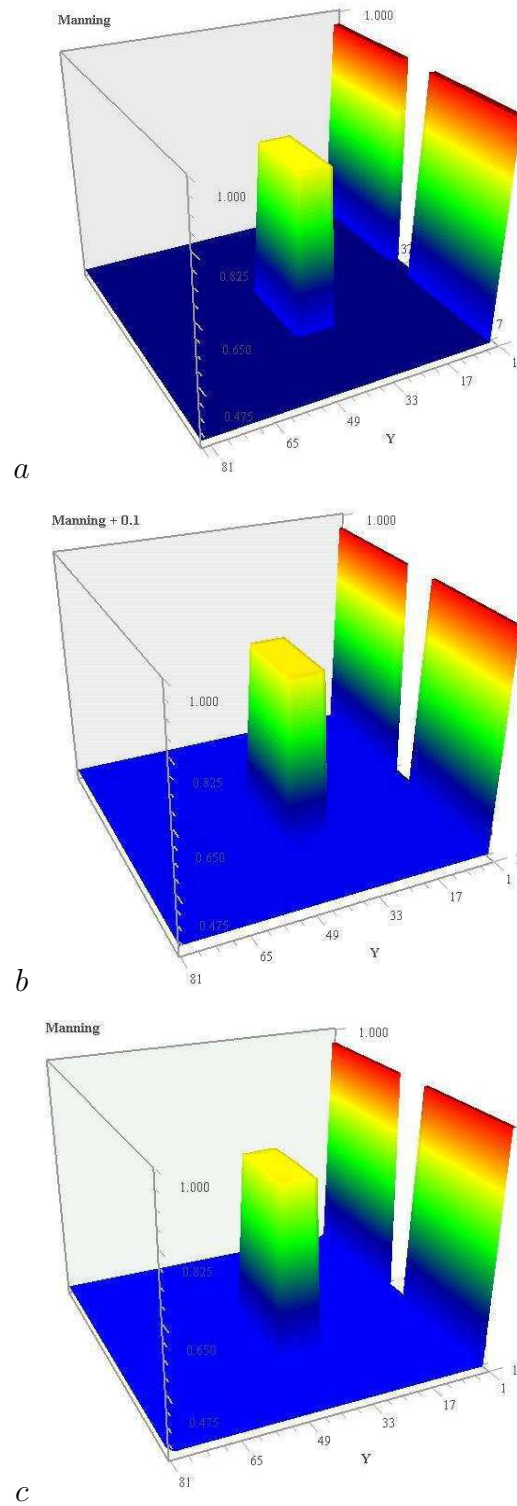
Soit un domaine carré ($50m \times 50m$) bordé à l'ouest par une digue. Le fond de ce bassin est plat. Au milieu de la digue se trouve une brèche par laquelle l'eau s'infiltre dans le bassin. La condition aux limites ouest sera donc définie grâce à des couples (hauteur, temps).

Les conditions aux limites nord et sud sont de type glissement sans frottement :

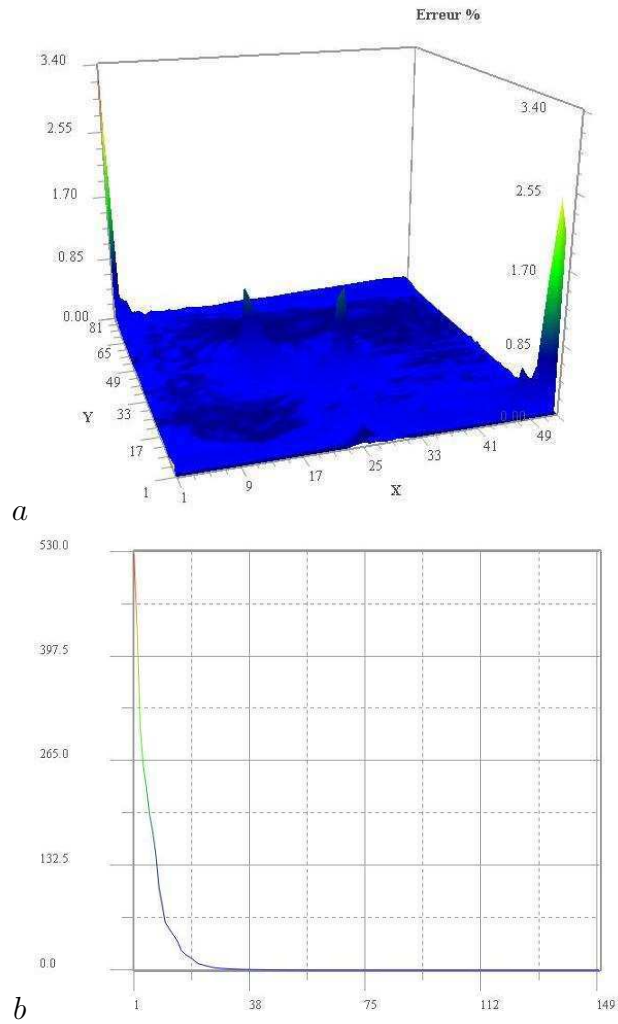
$$\vec{u} \cdot \vec{n} = 0$$

et la condition aux limites est de type frontière ouverte

$$\frac{\partial \vec{u}}{\partial \vec{n}} = 0$$



TAB. 6.5 – Illustration de l'estimation de la rugosité pour le cas test de la colline. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est celle que l'assimilation de données variationnelle doit nous aider à retrouver. b- La rugosité permettant de démarrer l'exécution du code d'optimisation, celle-ci est assez éloignée de celle que nous devons retrouver. c- La rugosité obtenue grâce à la minimisation de la fonction coût.



TAB. 6.6 – *Illustration de l'efficacité de l'estimation de la rugosité pour le cas test de la colline. a- La différence entre le jeu de rugosité initial et celui obtenu grâce à la minimisation de la fonction coût.. b- Evolution de la valeur de la fonction coût (6.1) en fonction des itérations du code d'optimisation.*

Le temps de l'écoulement sera de 1000 secondes avec un pas de temps de 2 secondes.

Le jeu de Manning, représentant la nature du sol, est très hétérogène (voir la figure a dans le tableau 6.7).

Le but de cette étude sera de montrer que l'estimation de ce paramètre est possible grâce à un jeu d'observations sur la zone, fourni par la méthode des expériences jumelles, les résultats du modèle direct et la minimisation d'une fonction coût par la méthode de l'assimilation de données variationnelle. La fonction coût utilisée dans cet exemple est

$$J(Manning) = \frac{1}{2}\|U - U_{obs}\|^2 + \frac{1}{2}\|V - V_{obs}\|^2 + \frac{1}{2}\|Zn - Zn_{obs}\|^2 \quad (6.4)$$

où (U, V) sont les coordonnées de la vitesse, Zn la hauteur d'eau et X_{obs} les observations fournies par la méthode des expériences jumelles (avec $X = \{U, V, Zn\}$). Les observations sont disponibles au temps $t = 1000$.

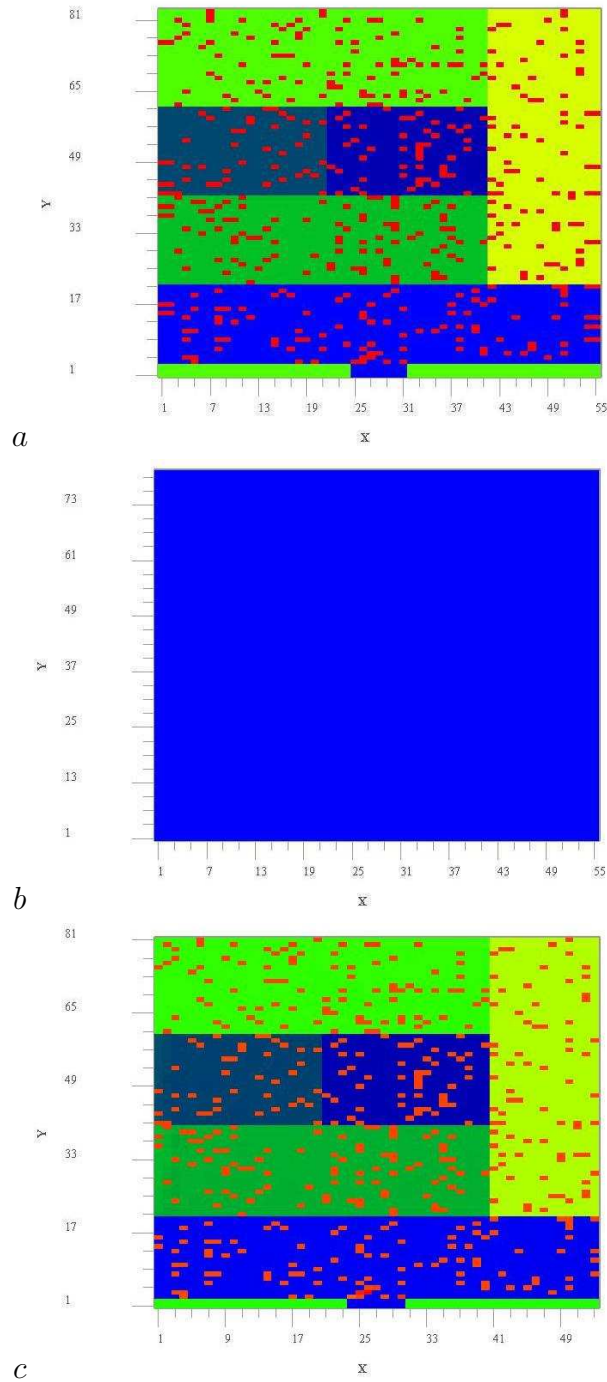
Estimation du jeu de Manning original

Avec le domaine prédéfini précédemment, nous allons mener dans un premier temps une estimation du jeu de Manning point par point.

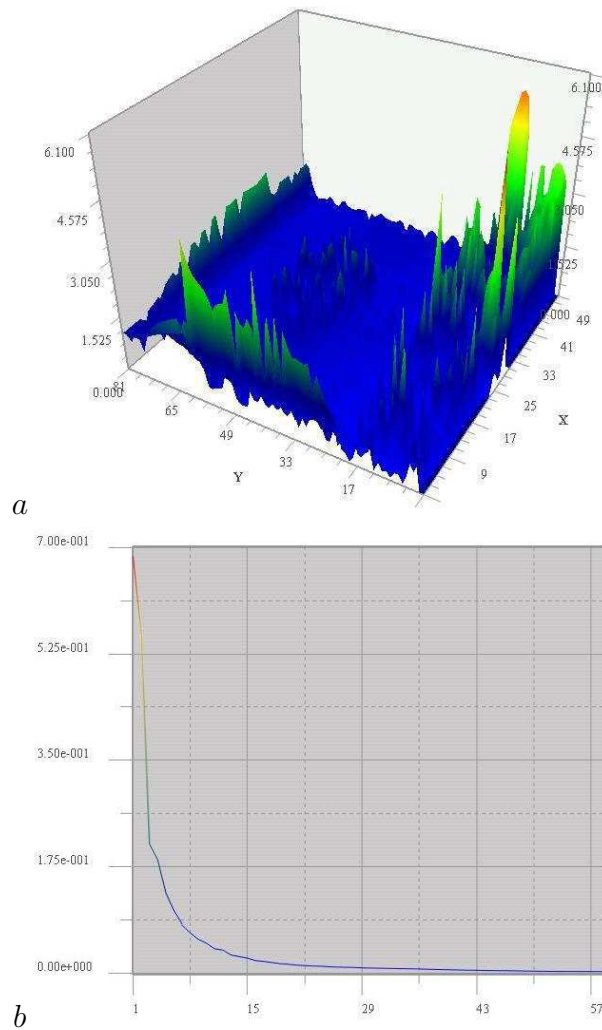
Nous initialisons l'exécution du code d'optimisation avec un jeu de Manning uniformément égal sur tout le domaine (voir la figure b dans le tableau 6.7).

Le tableau 6.7 dans cette section démontre l'efficacité de l'estimation du Manning, malgré la très grande différence entre le jeu à retrouver et le jeu utilisé pour initialiser l'exécution du code d'optimisation. L'erreur ainsi commise est insignifiante puisque inférieure à 3% dans le bassin (voir le tableau 6.8).

Lorsque l'on regarde précisément la construction du jeu de Manning à retrouver, on peut remarquer qu'il se découpe en 6 zones distinctes (voir la figure a dans le tableau 6.8), de Manning globalement identiques. Ces 6 zones sont légèrement bruitées par des Mannings épars. On peut se demander s'il est vraiment utile de tenir compte de tous ces coefficients isolés. Est-il alors possible de découper réellement ce bassin en six zones de Mannings identiques tout en conservant le même comportement du flux ?



TAB. 6.7 – *Illustration de l'estimation de la rugosité pour un jeu de Manning très hétérogène. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est celle que l'assimilation de données variationnelle doit nous aider à retrouver. b- La rugosité permettant de démarrer l'exécution du code d'optimisation, celle-ci est assez éloignée de celle que nous devons retrouver. c- La rugosité obtenue grâce à la minimisation de la fonction coût.*



TAB. 6.8 – Illustration de l'efficacité de l'estimation de la rugosité pour un jeu de Manning très hétérogène. a- La différence entre le jeu de rugosité initial et celui obtenu grâce à la minimisation de la fonction coût.. b- Evolution de la valeur de la fonction coût (6.1) en fonction des itérations du code d'optimisation.

Estimation du Manning par zones

Nous allons donc utiliser les observations fournies par le code direct grâce à la méthode des expériences jumelles à partir du jeu de Manning initial. Et nous considérerons un domaine où les Mannings seront définis uniquement par zones. Les zones étant déterminées *a priori*.

La minimisation de la fonction coût (6.4) permet une convergence vers 6665 (cf figure 6.1). Il est logique de ne pas avoir une convergence vers 0, puisque nous ne cherchons pas le jeu de Manning exact mais plutôt une schématisation de la donnée.

L'exécution du code adjoint nous permet d'obtenir un jeu de 6 Mannings (cf figure c dans le tableau 6.9) dont la donnée permet d'être relativement proche du flux attendu.

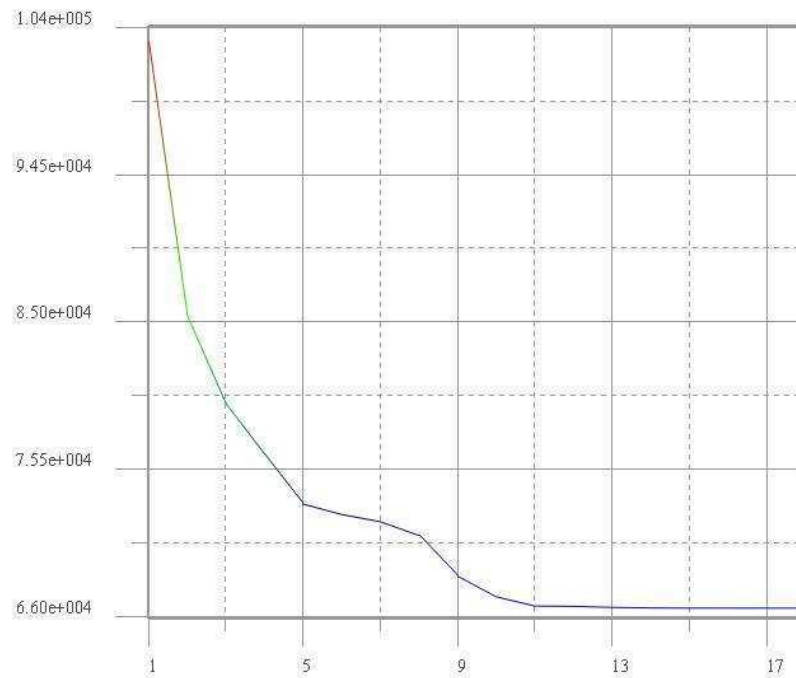
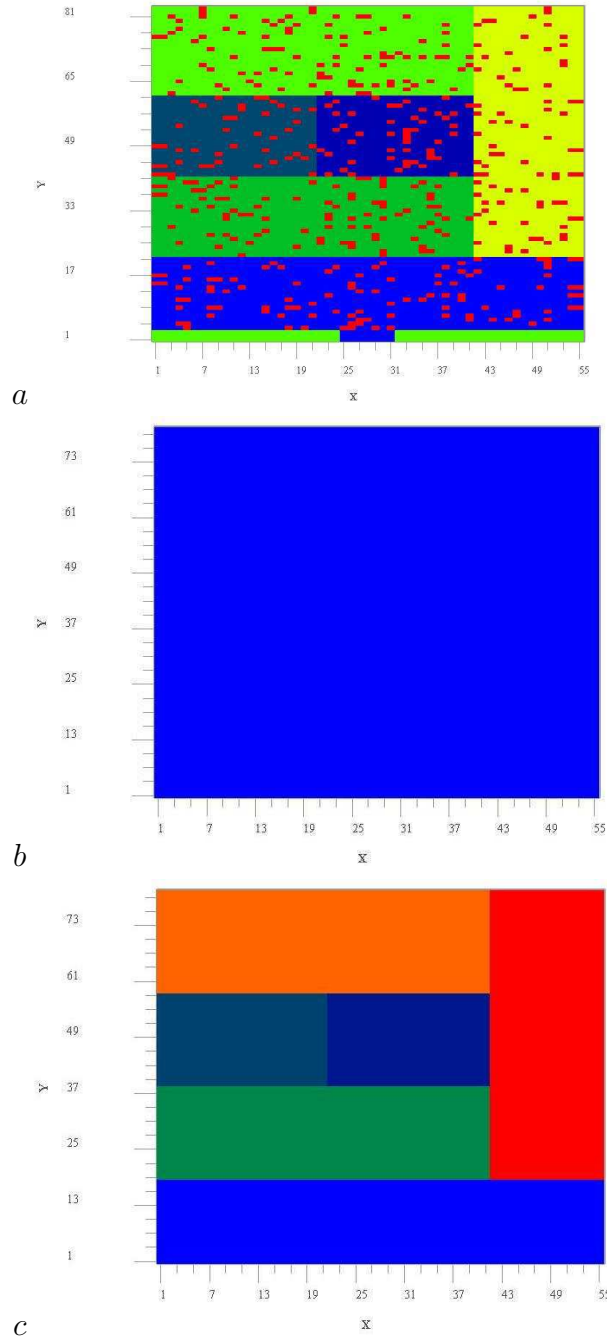


FIG. 6.1 – Evolution de la valeur de la fonction coût (6.4) en fonction des itérations du code d'optimisation.

En hydraulique opérationnelle les observations ne sont faites que sur les hauteurs d'eau. Nous allons voir dans la suite comment se comporte la minimisation de la fonction coût dans le cas où celle-ci ne mesure que la différence entre résultats et observations sur les hauteurs d'eau.



TAB. 6.9 – Illustration de l'estimation de la rugosité par zone. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est celle que l'assimilation de données variationnelle doit nous aider à approcher. b- La rugosité permettant de démarrer l'exécution du code d'optimisation, celle-ci est assez éloignée de celle que nous devons retrouver. c- La rugosité obtenue grâce à la minimisation de la fonction coût.

Observations sur les hauteurs d'eau

La fonction coût utilisée dans cet exemple mesure seulement la différence entre les résultats du code direct et les observations sur les hauteurs d'eau :

$$J(Manning) = \frac{1}{2} \|Z_n - Z_{n_{obs}}\|^2 \quad (6.5)$$

La convergence de la fonction coût vers 200 est présentée dans la figure 6.2. Cette convergence assure une bonne qualité du comportement du flux sur les hauteurs d'eau.

Le jeu de Manning obtenu (voir figure c dans le tableau 6.10) étant très différent de celui trouvé précédemment, il est clair que la vitesse d'écoulement calculée n'est pas exactement la même que celle attendue, mais l'erreur relative commise étant très faible le résultat est tout de même acceptable.

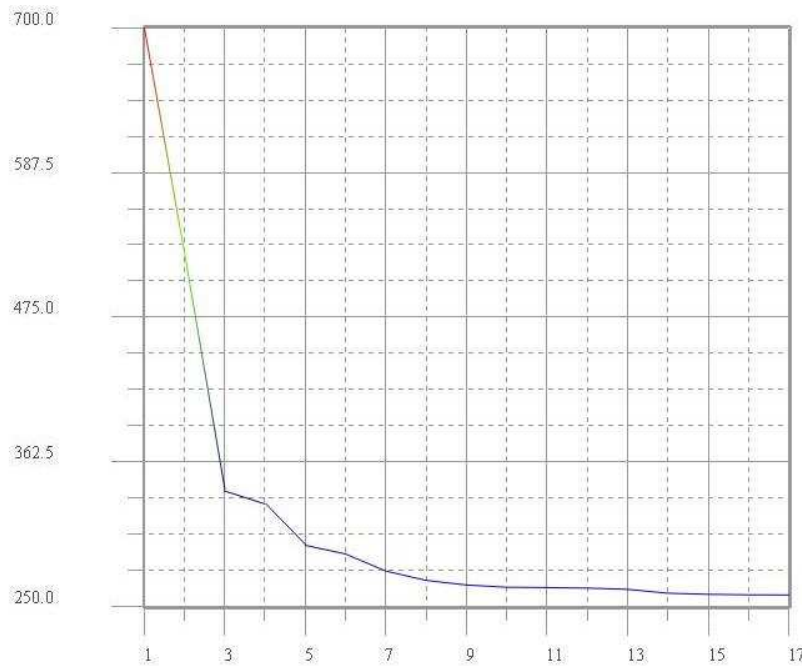
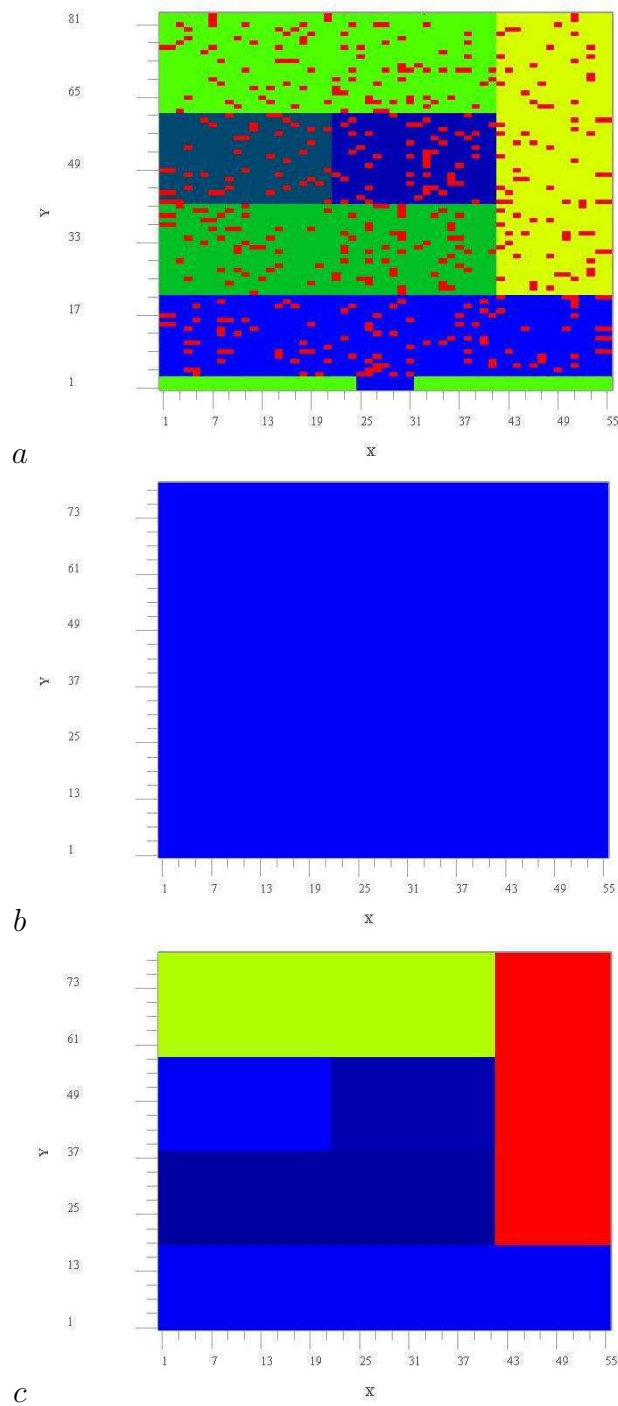


FIG. 6.2 – Dans la colonne Fonction coût : C'est l'évolution de la valeur de la fonction coût (6.5) en fonction des itérations du code d'optimisation.

Il n'est pas forcément judicieux d'utiliser toutes les informations du jeu de Manning afin de modéliser un domaine. La schématisation de cette donnée illustrée précédemment peut être menée de manière rigoureuse et fournir des résultats encourageants. En effet, le découpage du domaine précédent en zones de Manning équivalents permet une bonne reconstitution de l'écoulement de l'eau tout en simplifiant l'entrée de la donnée rugosité.

La distribution initiale du jeu de rugosité nous a imposé ce découpage en six zones. Quels sont les critères de découpage possible dans le cas d'une distribution plus complexe ?



TAB. 6.10 – Illustration de l'estimation de la rugosité par zone avec une fonction coût mesurant l'erreur commise sur le calcul de la hauteur d'eau. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est celle que l'assimilation de données variationnelle doit nous aider à approcher. b- La rugosité permettant de démarrer l'exécution du code d'optimisation, celle-ci est assez éloignée de celle que nous devons retrouver. c- La rugosité obtenue grâce à la minimisation de la fonction coût.

6.1.5 Loire

Soit un domaine carré (50 m × 50 m) bordé à l'ouest par une digue. Au milieu de la digue se trouve une brèche par laquelle l'eau s'infiltre dans le bassin. La condition aux limites ouest sera donc définie grâce à des couples (hauteur, temps).

Les conditions aux limites nord et sud sont de type glissement sans frottement :

$$\vec{u} \cdot \vec{n} = 0$$

et la condition aux limites est de type frontière ouverte

$$\frac{\partial \vec{u}}{\partial \vec{n}} = 0$$

Le temps de l'écoulement sera de 1000 secondes avec un pas de temps de 2 secondes.

Le domaine considéré est situé en aval d'Orléans dans le val d'Ouzouer dans la plaine inondable de la Loire (voir le carré rouge dans la figure a du tableau 6.11). Le relief de cette zone est obtenu par photogrammétrie² et le jeu de Manning a été déduit d'une image SPOT³ (voir la figure c dans le tableau 6.11).

Le but de cette étude sera de trouver quelques critères pour découper le domaine en zones où le Manning sera estimé de manière globale afin de montrer que la schématisation de ce paramètre est possible grâce à la minimisation d'une fonction coût. La fonction coût utilisée dans cet exemple est

$$J(Manning) = \frac{1}{2} \|U - U_{obs}\|^2 + \frac{1}{2} \|V - V_{obs}\|^2 + \frac{1}{2} \|Zn - Zn_{obs}\|^2 \quad (6.6)$$

où (U, V) sont les coordonnées de la vitesse, Zn la hauteur d'eau et X_{obs} les observations fournies par la méthode des expériences jumelles (avec $X = U, V, Zn$). Les observations sont disponibles au temps $t = 1000$.

Trouver un critère afin de découper le domaine en zones de Manning équivalents, c'est remarquer un phénomène qui pourrait modifier le comportement de l'écoulement de l'eau. *A priori*, le paramètre qui va le plus influencer l'écoulement de l'eau est la topographie. Nous allons donc étudier ci-après deux critères de sélection liés au relief.

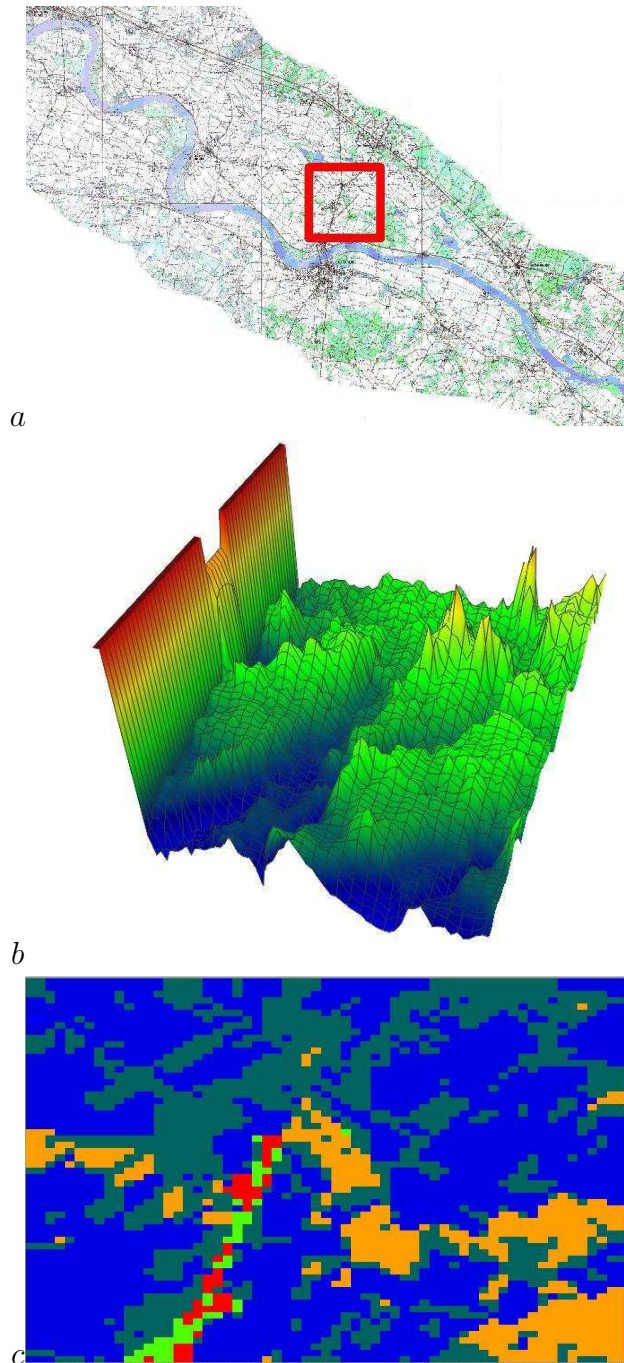
Découpage suivant l'altitude

Dans un premier temps, nous allons essayer de découper le domaine en fonction de l'altitude du relief en chaque point de la grille. Le découpage se fera en 10 zones entre les altitudes 100 m et 108 m.

Hauteur	.< 100	100 < .< 101	101 < .< 102	102 < .< 103	103 < .< 104
Zone	1	2	3	4	5
Hauteur	104 < .< 105	105 < .< 106	106 < .< 107	107 < .< 108	.> 108
Zone	6	7	8	9	10

²La précision ainsi obtenue est de 20 cm en altitude. On a un point tous les 20 m.

³De l'image SPOT une classification de l'occupation des zones a été extraite. Cette classification se décompose en 5 parties (forêt, ville, zones urbanisées, champ et zones aquatiques).



TAB. 6.11 – *Présentation du domaine Loire. a- Représentation du Val d'Ouzouer en amont d'Orléans sur la Loire. La zone entourée de rouge est le domaine considéré dans les exemples présentés ci dessous. b- Le relief d'une partie du Val d'Ouzouer sur la Loire. c- La représentation de la rugosité sur une partie du val d'Ouzouer (SPOT).*

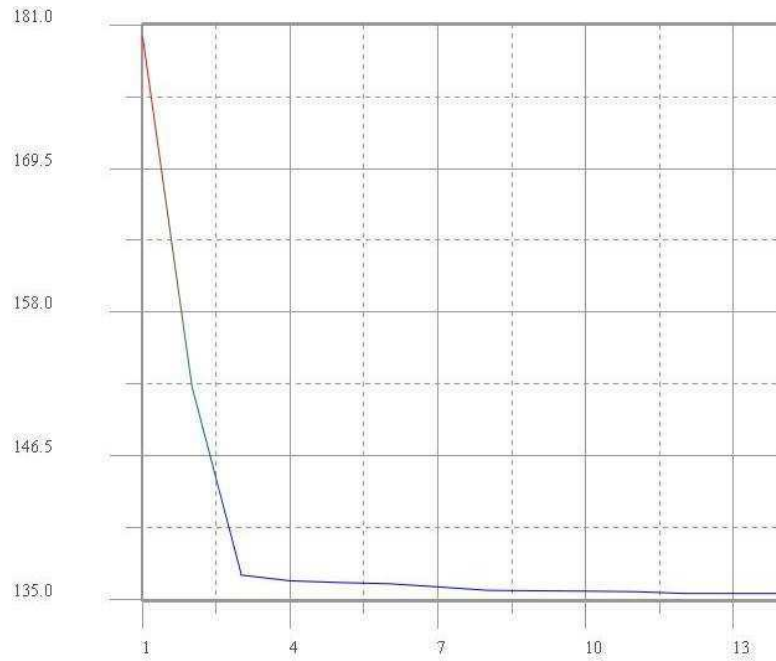


FIG. 6.3 – Evolution de la valeur de la fonction coût (6.6) en fonction des itérations du code d'optimisation.

Cette subdivision est complètement arbitraire et ne relève pas de considérations physiques.

Après initialisation du jeu de Manning de manière aléatoire, la minimisation de la fonction coût (voir la figure 6.3) nous permet d'obtenir un jeu de Manning par zones, qui n'a plus rien à voir avec le jeu de Manning initial, mais qui nous permet de conserver un écoulement pratiquement identique. En effet, le minimum de la fonction coût est atteint pour $J(\text{Manning}) = 200$.

Du point de vue du résultat, ce critère paraît intéressant. Dans la sous section suivante, nous allons en essayer un autre, basé non plus sur l'altitude à chaque point de la grille, mais plutôt sur la pente en ces mêmes points.

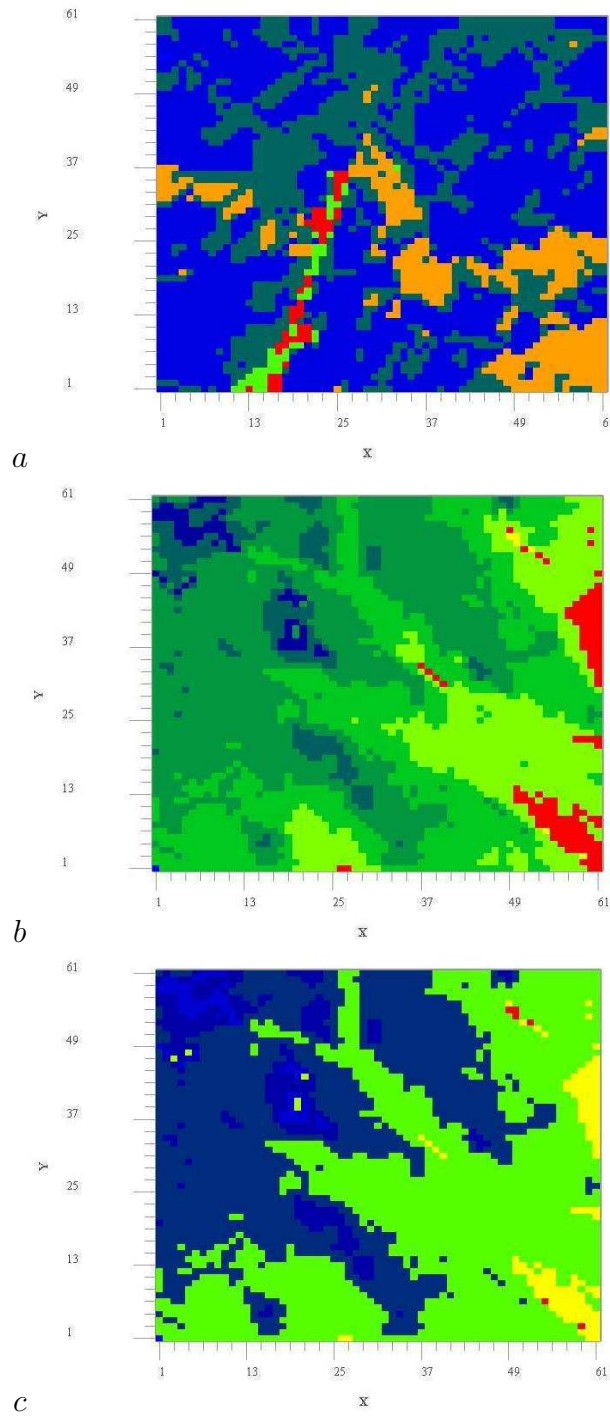
Découpage suivant la pente

Le découpage se fera donc en 10 zones en fonction du gradient de la pente.

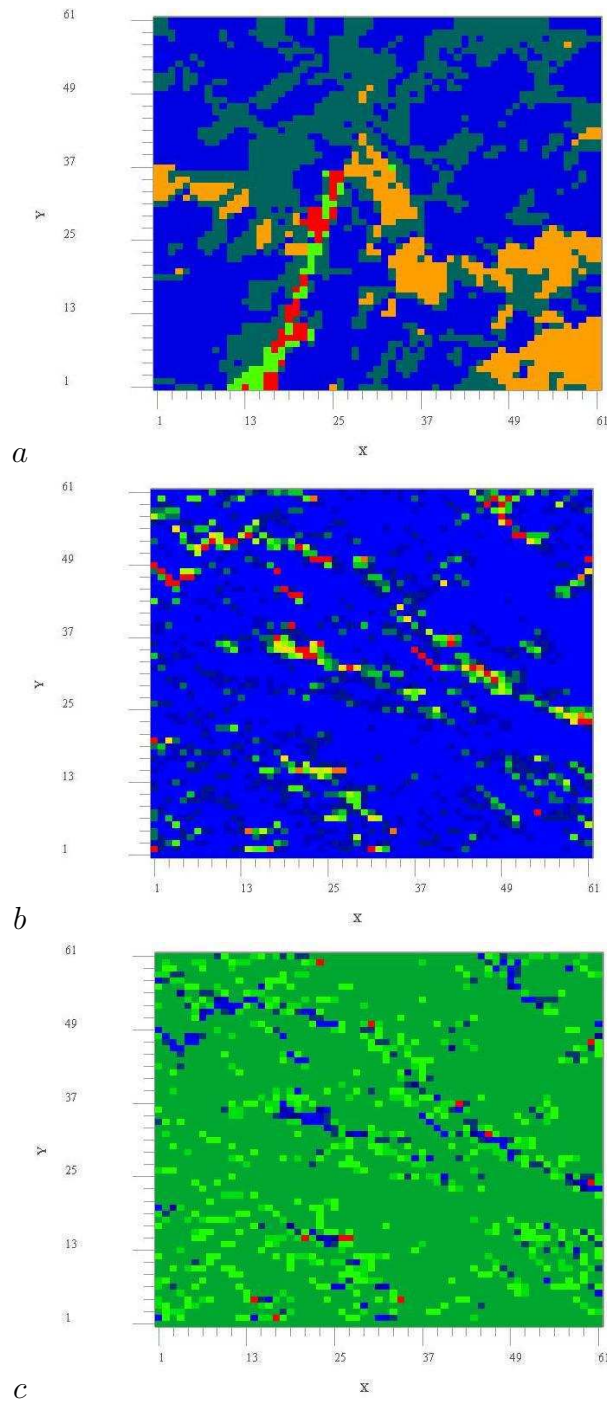
Hauteur	. < 0.1	0.1 < . < 0.2	0.2 < . < 0.3	0.3 < . < 0.4	0.4 < . < 0.5
Zone	1	2	3	4	5
Hauteur	0.5 < . < 0.6	0.6 < . < 0.7	0.7 < . < 0.8	0.8 < . < 0.9	. > 0.9
Zone	6	7	8	9	10

Cette subdivision est complètement arbitraire et ne relève pas de considérations physiques.

Après initialisation du jeu de Manning de manière aléatoire, comme précédemment, la minimisation de la fonction coût (voir figure 6.4) nous permet d'obtenir un jeu de Manning par zones, qui n'a plus rien à voir avec le jeu de Manning initial. Le minimum de la fonction



TAB. 6.12 – *Illustration de l'estimation de la rugosité par zone pour le domaine Loire grâce à un critère de découpage fonction de l'altitude. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est celle que l'assimilation de données variationnelle doit nous aider à approcher. b- La rugosité permettant de démarrer l'exécution du code d'optimisation, celle-ci est assez éloignée de celle que nous devons retrouver. c- La rugosité obtenue grâce à la minimisation de la fonction coût.*



TAB. 6.13 – *Illustration de l'estimation de la rugosité par zone pour le domaine Loire grâce à un critère de découpage fonction de la pente. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est celle que l'assimilation de données variationnelle doit nous aider à approcher. b- La rugosité permettant de démarrer l'exécution du code d'optimisation, celle-ci est assez éloignée de celle que nous devons retrouver. c- La rugosité obtenue grâce à la minimisation de la fonction coût.*

coût est atteint pour $J(Manning) = 190$, ce qui est pratiquement identique au résultat trouvé dans la sous-section antérieure.

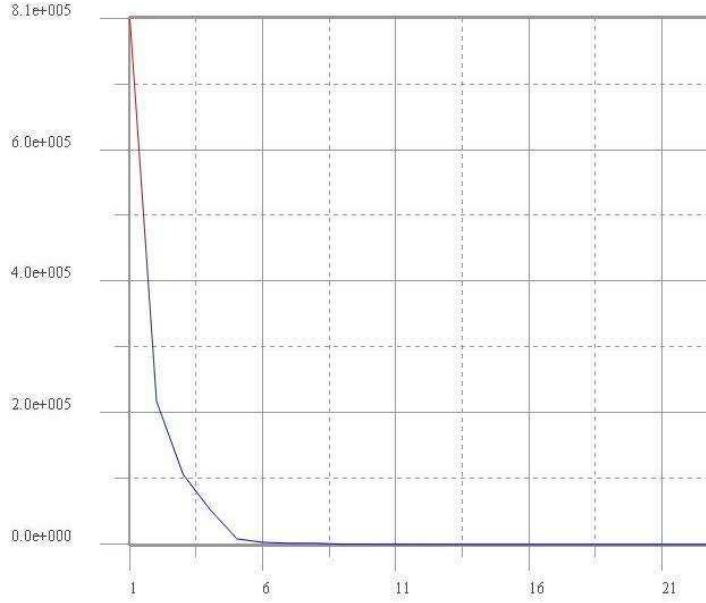


FIG. 6.4 – Dans la colonne *Fonction coût* : C'est l'évolution de la valeur de la fonction coût (6.6) en fonction des itérations du code d'optimisation.

Les deux exemples sur le choix du critère de découpage du domaine étudié précédemment nous donnent des résultats équivalents. Bien que la répartition spatiale des rugosité n'est plus du tout la même, ces résultats sont de bonne qualité et semblent valider les critères utilisés. Mais ces critères n'ont aucune valeur scientifique puisqu'ils ont été choisis de manière aléatoire. Nous verrons dans la suite l'utilisation de l'analyse de sensibilité, introduite dans le premier chapitre, comme critère de découpage.

6.2 Estimation du relief équivalent

6.2.1 La colline

Pour débiter cette étude sur l'estimation du relief équivalent, nous allons mettre en place un exemple avec le logiciel LMCFLD.

Soit un domaine carré ($50m \times 50m$) bordé à l'ouest par une digue. Au milieu de la digue se trouve une brèche par laquelle l'eau s'infiltre dans le bassin. La condition aux limites ouest sera donc définie grâce à des couples (hauteur, temps).

Les conditions aux limites nord et sud sont de type glissement sans frottement :

$$\vec{u} \cdot \vec{n} = 0$$

et la condition aux limites est de type frontière ouverte

$$\frac{\partial \vec{u}}{\partial \vec{n}} = 0$$

Le temps de l'écoulement sera de 1000 secondes avec un pas de temps de 2 secondes.

Le jeu de Manning, représentant la nature du sol, est uniformément égal à 0.05.

Le fond du bassin est plat excepté au centre où se trouve une zone surélevée, cet exemple pourrait tout à fait simuler une colline au milieu d'une zone de moindre rugosité.

Le but de cette étude sera de montrer que l'estimation du relief équivalent est possible grâce à un jeu d'observations sur la zone, fourni par la méthode des expériences jumelles.

Avec un jeu d'observations complet

La fonction coût à minimiser dans cet exemple est

$$J(Z_b) = \frac{1}{2}\|U - U_{obs}\|^2 + \frac{1}{2}\|V - V_{obs}\|^2 + \frac{1}{2}\|Z_n - Z_{n_{obs}}\|^2 \quad (6.7)$$

où (U, V) sont les coordonnées de la vitesse, Z_n la hauteur d'eau et X_{obs} les observations fournies par la méthode des expériences jumelles (avec $X = \{U, V, Z_n\}$) et Z_b la topographie. Les observations sont disponibles au temps $t = 1000$.

Pour initialiser le code d'optimisation, nous allons utiliser une topographie où la colline a été effacée (voir la figure b du tableau 6.14). Le but de cette estimation va être de voir à quel point on peut la retrouver.

Après quelques itérations du code d'optimisation, le minimum de la fonction coût a été atteint. Cette minimisation nous a permis d'extraire une topographie à l'optimum. On peut voir dans le premier tableau que la colline a bien été retrouvée avec une erreur minime (voir tableau précédent).

Ce bon résultat a été obtenu grâce à un jeu d'observations complet sur toute la zone, mais qu'en aurait-il été si le jeu d'observations avait été moins dense ?

Avec un jeu d'observations sur les hauteurs d'eau

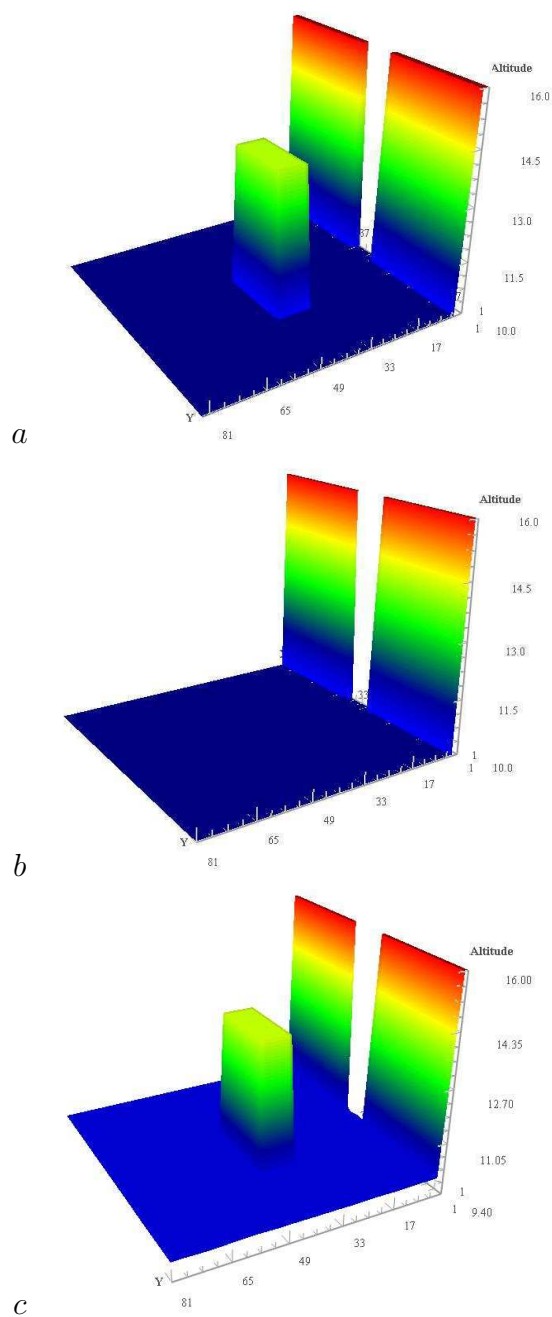
Dans cet exemple, nous allons limiter les observations à la donnée portant sur les hauteurs d'eau seulement. Ainsi la fonction coût à minimiser dans cet exemple est

$$J(Z_b) = \frac{1}{2}\|Z_n - Z_{n_{obs}}\|^2 \quad (6.8)$$

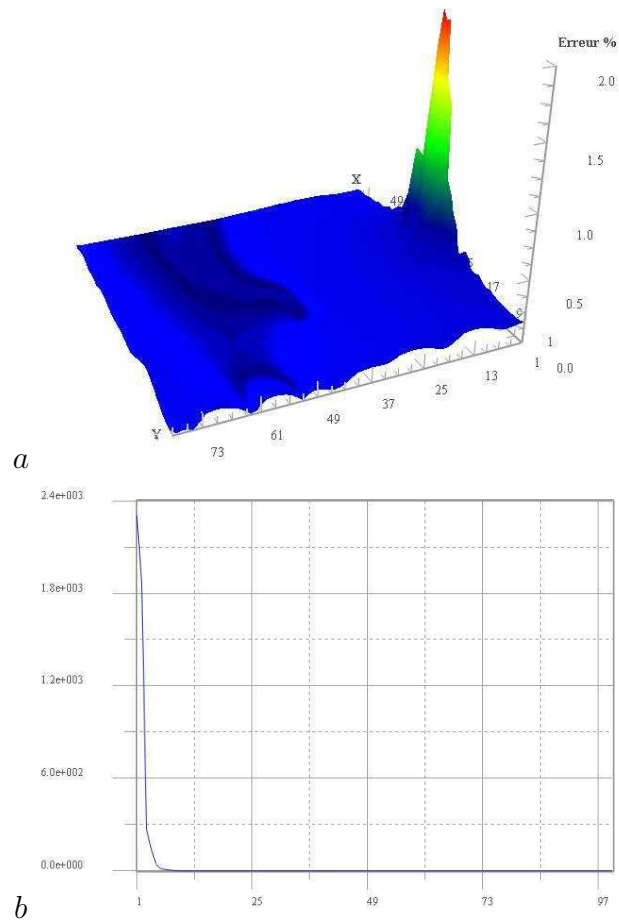
où Z_n est la hauteur d'eau, $Z_{n_{obs}}$ les observations fournies par la méthode des expériences jumelles et Z_b la topographie. Les observations sont disponibles au temps $t = 1000$.

La topographie pour initialiser l'exécution du code d'optimisation est la même que celle utilisée dans l'exemple précédent (cf tableau 6.16).

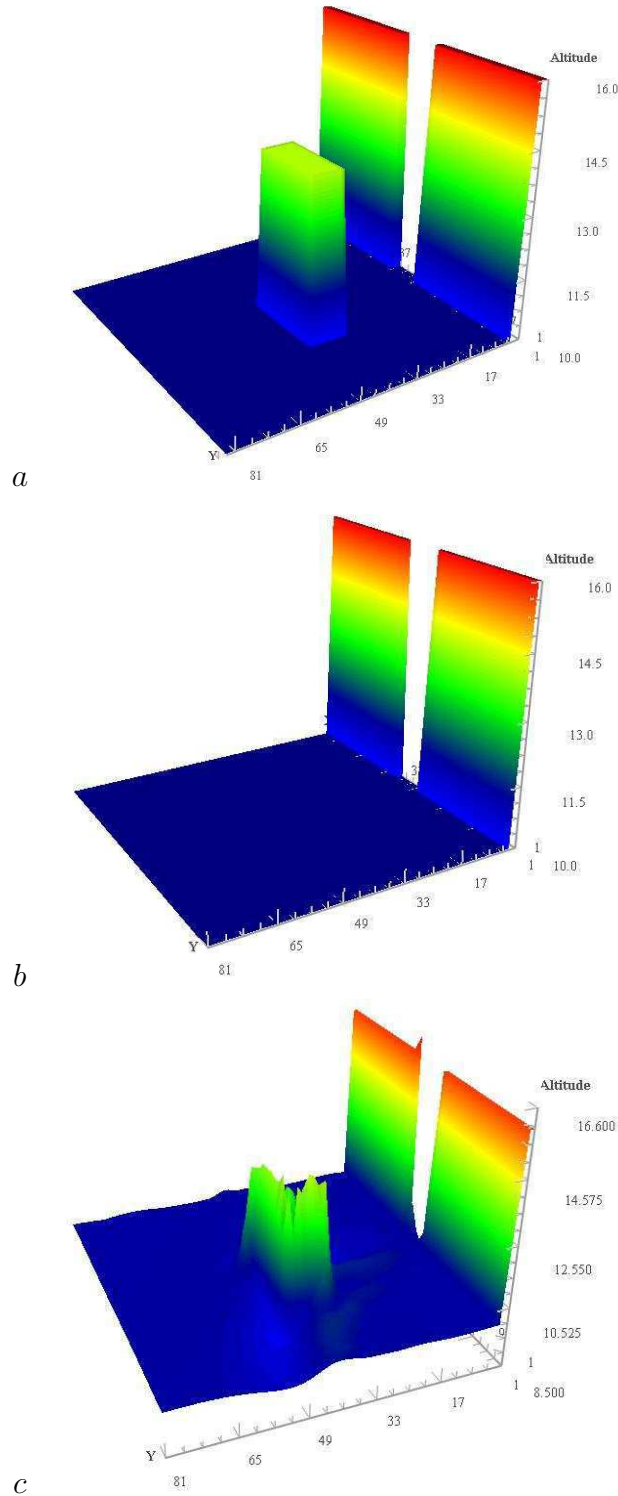
La dégradation du jeu d'observations a quelque peu dégradé la qualité de la topographie équivalente obtenue. En effet, la position et la forme de la base de la colline ont bien été retrouvées (cf figure a dans le tableau 6.17). Cependant, la taille de cette colline n'a pas été parfaitement trouvée puisque l'erreur moyenne commise est de 15% à l'emplacement de celle-ci.



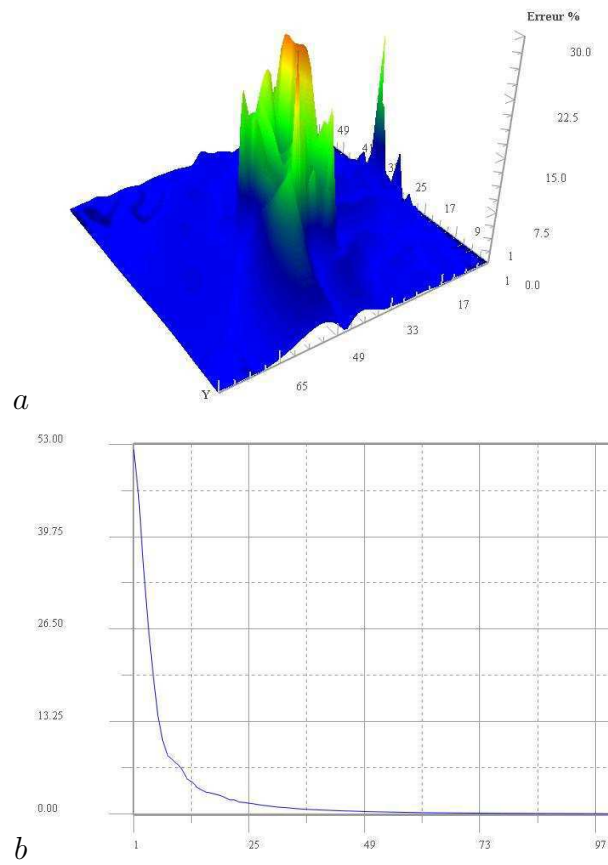
TAB. 6.14 – Illustration de l'estimation de la topographie équivalente pour le cas test de la colline. *a*- La topographie du bassin. Ce relief est celui que l'assimilation de données variationnelle doit nous aider à retrouver. *b*- Le relief permettant de démarrer l'exécution du code d'optimisation, celui-ci est assez éloigné de celui que nous devons retrouver. *c*- La topographie obtenue grâce à l'aide de la minimisation de la fonction coût.



TAB. 6.15 – Illustration de l'efficacité de l'estimation de la topographie équivalente pour le cas test de la colline. a- La différence entre la topographie initiale et celle obtenu grâce à la minimisation de la fonction coût. b- Evolution de la valeur de la fonction coût (6.7) en fonction des itérations du code d'optimisation.



TAB. 6.16 – Illustration de l'estimation de la topographie équivalente pour le cas test de la colline avec une fonction coût ne mesurant l'erreur que sur les hauteurs d'eau. a- La topographie du bassin. Ce relief est celui que l'assimilation de données variationnelle doit nous aider à retrouver. b- Le relief permettant de démarrer l'exécution du code d'optimisation, celui-ci est assez éloigné de celui que nous devons retrouver. c- La topographie obtenue grâce à l'aide de la minimisation de la fonction coût.



TAB. 6.17 – *Illustration de l’efficacité de l’estimation de la topographie équivalente pour le cas test de la colline. a- La différence entre la topographie initiale et celle obtenu grâce à la minimisation de la fonction coût. b- Evolution de la valeur de la fonction coût (6.7) en fonction des itérations du code d’optimisation.*

6.2.2 La route

Soit un domaine carré ($50m \times 50m$) bordé à l'ouest par une digue. Au milieu de la digue se trouve une brèche par laquelle l'eau s'infiltre dans le bassin. La condition aux limites ouest sera donc définie grâce à des couples (hauteur, temps).

Les conditions aux limites nord et sud sont de type glissement sans frottement :

$$\vec{u} \cdot \vec{n} = 0$$

et la condition aux limites est de type frontière ouverte

$$\frac{\partial \vec{u}}{\partial \vec{n}} = 0$$

Le temps de l'écoulement sera de 1000 secondes avec un pas de temps de 2 secondes.

Le jeu de Manning, représentant la nature du sol, est uniformément égal à 0.05.

Le fond du bassin est plat excepté une zone surélevée qui traverse le bassin en son milieu de part en part, cet exemple simule une route traversant une zone de moindre rugosité.

Le but de cette étude sera de montrer que l'estimation du relief équivalent est possible grâce à un jeu d'observations, fourni par la méthode des expériences jumelles. Les observations sont disponibles au temps $t = 1000$.

La fonction coût à minimiser dans cet exemple est :

$$J(Z_b) = \frac{1}{2} \|U - U_{obs}\|^2 + \frac{1}{2} \|V - V_{obs}\|^2 + \frac{1}{2} \|Z_n - Z_{n_{obs}}\|^2 \quad (6.9)$$

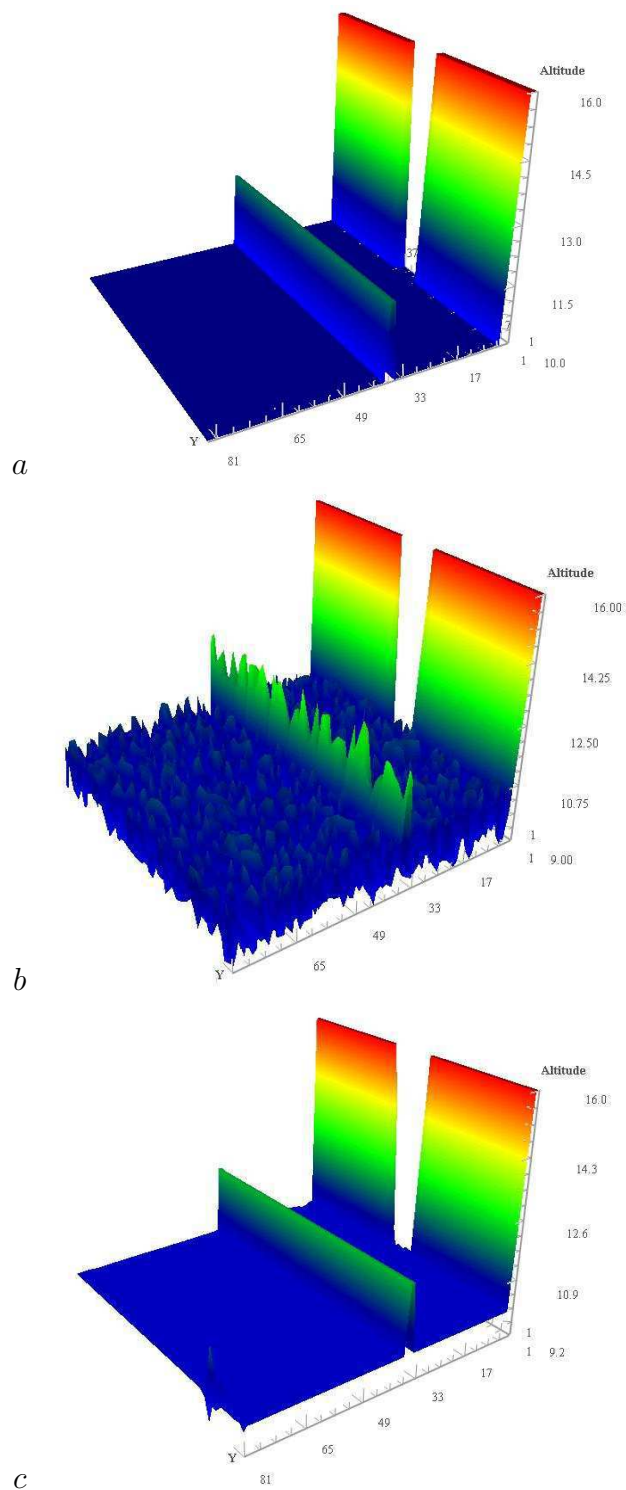
où (U, V) sont les coordonnées de la vitesse, Z_n la hauteur d'eau et X_{obs} les observations fournies par la méthode des expériences jumelles (avec $X = \{U, V, Z_n\}$).

Avec un jeu d'observations complet

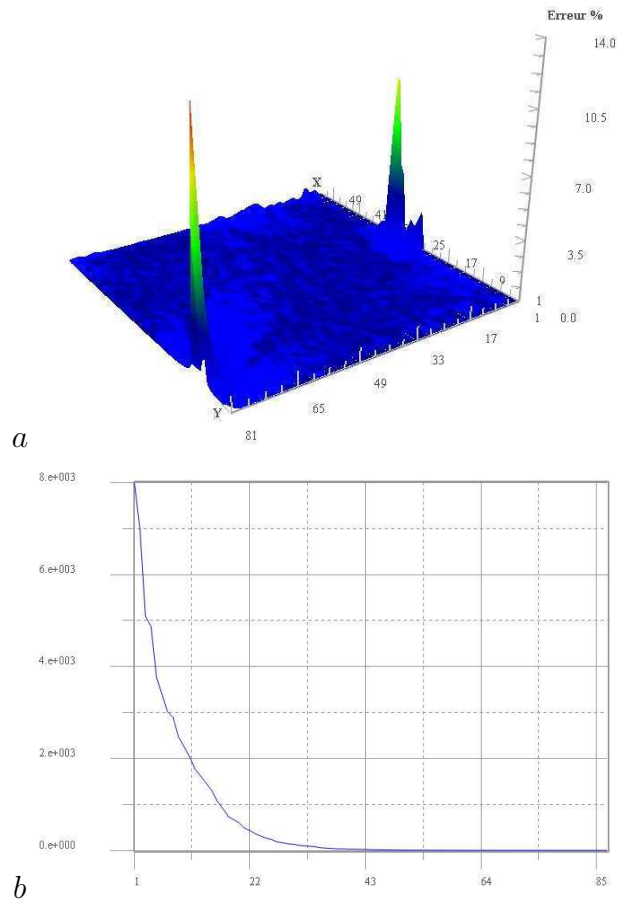
La topographie utilisée pour initialiser l'exécution du code d'optimisation est une perturbation totale de la topographie optimale (cf figure b dans le tableau 6.18).

Comme pour les exemples précédents, la topographie équivalente est bien retrouvée (cf figure c dans le tableau 6.18). La minimisation rapide de la fonction coût (cf figure b dans le tableau 6.19) ne dégradant pas la qualité du résultat, on peut affirmer que lorsque le jeu d'observations est complet, les résultats à attendre ne peuvent être que bons.

Comme pour l'exemple de la colline, nous allons dégrader doucement le jeu d'observations. Mais au lieu de dégrader le jeu en ne conservant que les observations sur la hauteur d'eau, nous allons dégrader l'information sur la donnée en espace.



TAB. 6.18 – Illustration de l'estimation de la topographie équivalente pour le cas test de la route avec un jeu d'observations complet. a- La topographie du bassin. Ce relief est celui que l'assimilation de données variationnelle doit nous aider à retrouver. b- Le relief permettant de démarrer l'exécution du code d'optimisation, celui-ci est assez éloigné de celui que nous devons retrouver. c- La topographie obtenue grâce à l'aide de la minimisation de la fonction coût.



TAB. 6.19 – Illustration de l'efficacité de l'estimation de la topographie équivalente pour le cas test de la route avec un jeu d'observations complet. a- La différence entre la topographie initiale et celle obtenu grâce à la minimisation de la fonction coût. b- Evolution de la valeur de la fonction coût (6.9) en fonction des itérations du code d'optimisation.

Avec un jeu d'observations partout sauf au dessus de la route

Dans cet exemple, les observations sont disponibles partout sur la zone excepté juste au dessus de la route. La fonction coût reste donc la même que précédemment, le champ d'action est juste réduit.

Cette dégradation du jeu d'observations ne pénalise pratiquement pas la performance de la méthode (voir la figure c dans le tableau 6.20). En effet, la topographie est tout aussi bien retrouvée bien qu'un léger bourrelet peut se voir dans la représentation de l'erreur (voir la figure a dans le tableau 6.21).

L'erreur commise afin d'estimer la topographie équivalente est minimale (cf figure a dans le tableau 6.21).

Ce résultat n'est pas surprenant puisque la route influence l'évolution de l'écoulement, non seulement au dessus d'elle mais aussi autour. On peut se demander si la qualité du résultat obtenu n'est pas due uniquement aux observations faites autour de la route.

Avec un jeu d'observations partout sauf autour de la route

Cette fois-ci, nous allons utiliser les mêmes observations que précédemment (partout sauf au dessus de la route) et nous allons en plus éliminer les observations autour de la route, c'est-à-dire toutes les données à moins de deux mailles de la route (sachant que la route elle-même a deux mailles de largeur).

Cette fois, le résultat est pénalisé par ce jeu d'observations réduit. (cf figure c dans le tableau 6.22).

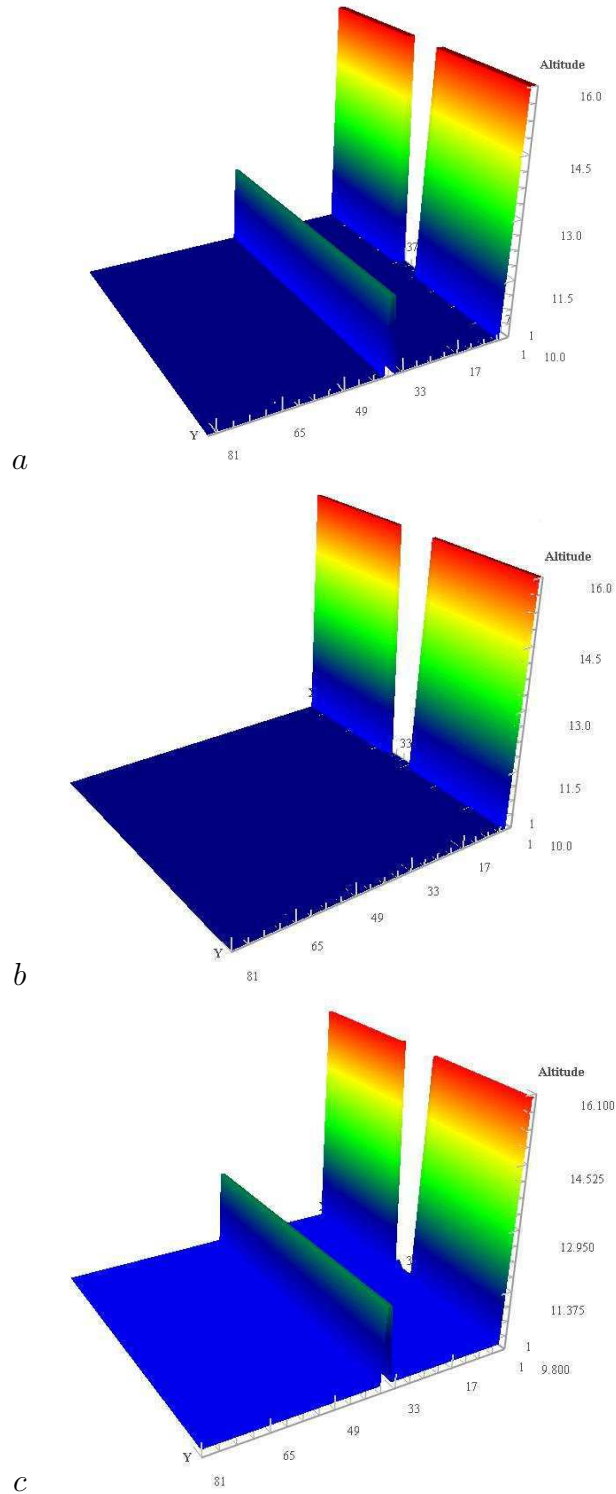
En effet, bien que la position de la route ainsi que sa largeur soient retrouvées, on perd totalement la notion de taille et l'erreur s'est agrandie (cf figure a dans le tableau 6.23). Ainsi, la méthode nous permet tout de même de retrouver la route mais pas dans sa forme optimale.

Avec un jeu d'observations seulement sur la route

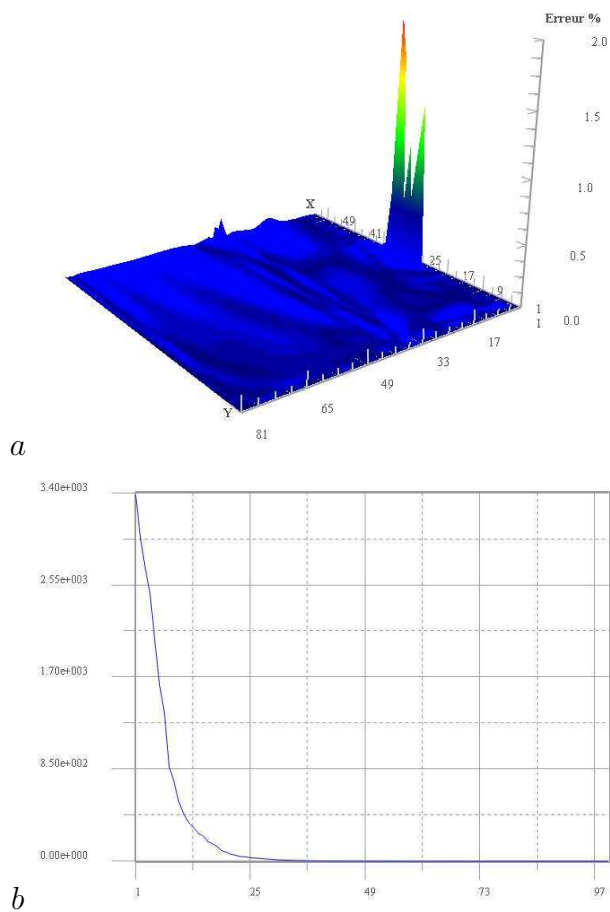
Dans ce dernier exemple, nous allons considérer le jeu d'observations à l'inverse de ce qui a été fait précédemment. En effet, nous allons supposer que les informations sont seulement disponibles au dessus de la route et nulle part ailleurs.

Les observations utilisées ici nous permettent de retrouver de manière satisfaisante la présence de la route (cf figure c dans le tableau 6.24). L'erreur est certes non négligeable autour de la route (cf figure a dans le tableau 6.25). Toutefois il est prouvé que l'on peut obtenir des résultats prometteurs à partir de très peu de données, mais suffisamment bien placées.

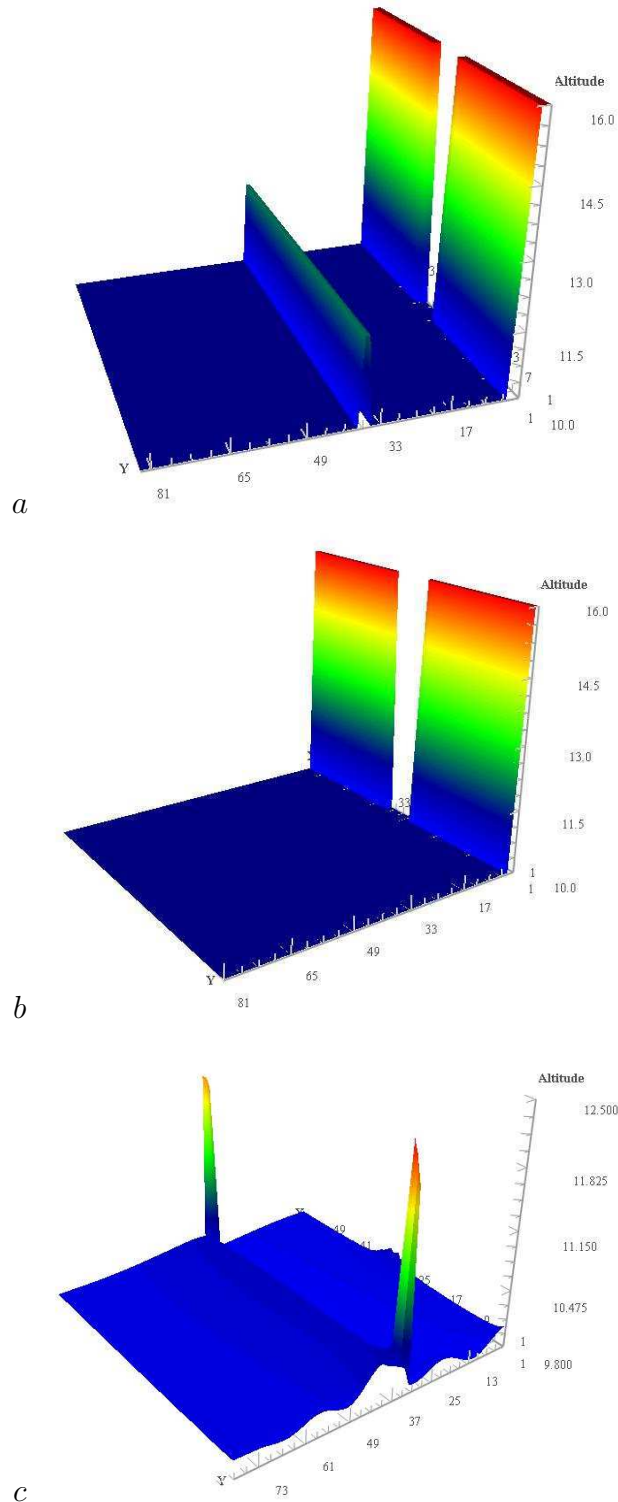
L'étude menée ici témoigne de l'importance du jeu d'observations utilisé. En effet, on a montré que la densité des observations n'est pas le seul critère à respecter mais que la position de ces observations est aussi essentielle.



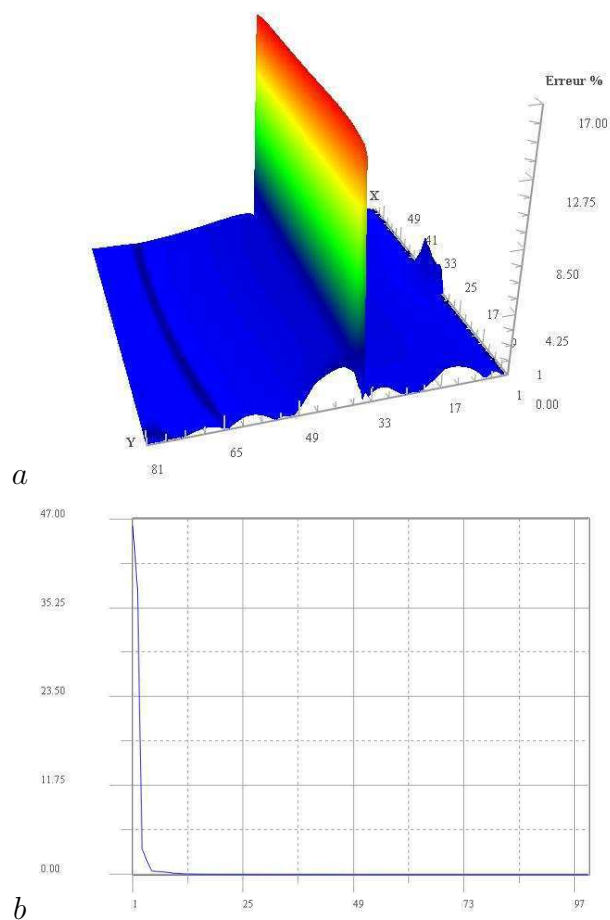
TAB. 6.20 – Illustration de l'estimation de la topographie équivalente pour le cas test de la route avec un jeu d'observations partout sauf au dessus de la route. a- La topographie du bassin. Ce relief est celui que l'assimilation de données variationnelle doit nous aider à retrouver. b- Le relief permettant de démarrer l'exécution du code d'optimisation, celui-ci est assez éloigné de celui que nous devons retrouver. c- La topographie obtenue grâce à l'aide de la minimisation de la fonction coût.



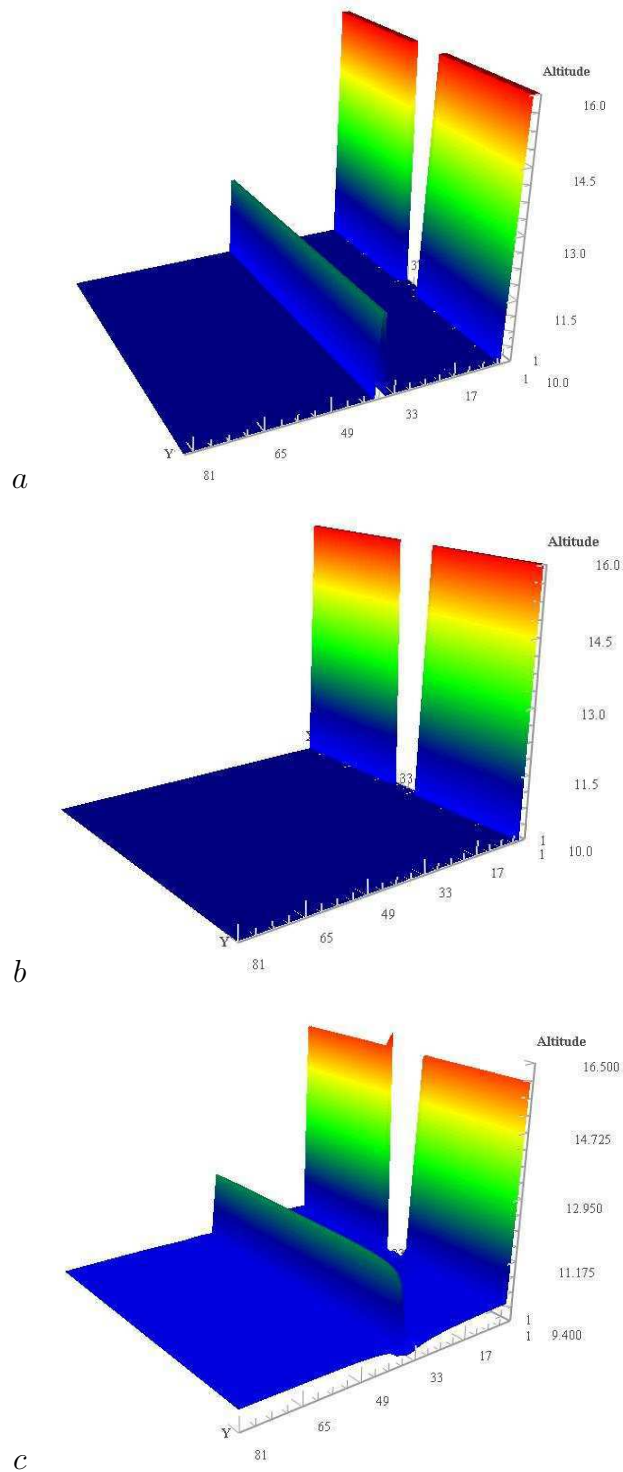
TAB. 6.21 – *Illustration de l'efficacité de l'estimation de la topographie équivalente pour le cas test de la route avec un jeu d'observations partout sauf au dessus de la route. a- La différence entre la topographie initiale et celle obtenu grâce à la minimisation de la fonction coût. b- Evolution de la valeur de la fonction coût (6.9) en fonction des itérations du code d'optimisation.*



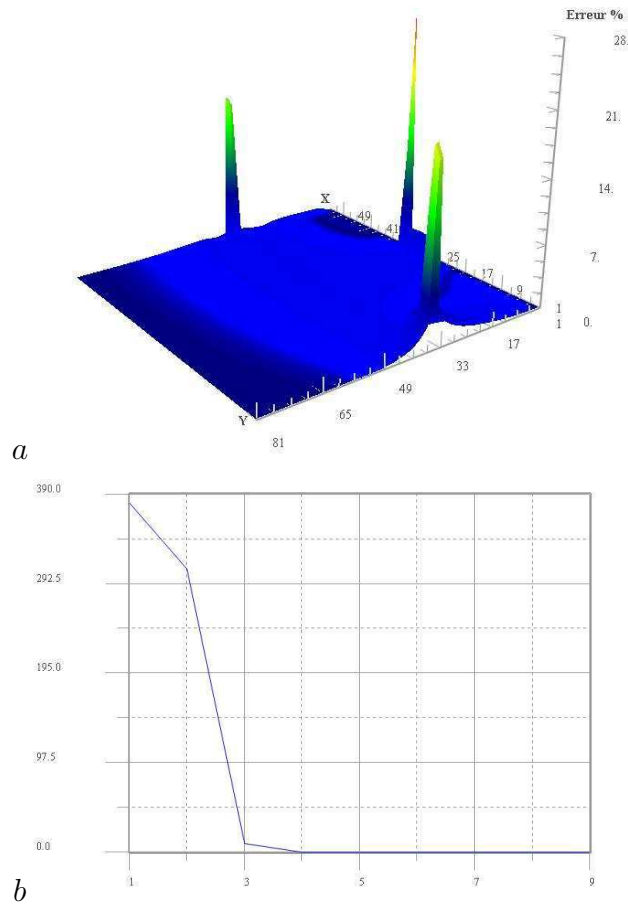
TAB. 6.22 – Illustration de l'estimation de la topographie équivalente pour le cas test de la route avec un jeu d'observations partout sauf autour de la route. a- La topographie du bassin. Ce relief est celui que l'assimilation de données variationnelle doit nous aider à retrouver. b- Le relief permettant de démarrer l'exécution du code d'optimisation, celui-ci est assez éloigné de celui que nous devons retrouver. c- La topographie obtenue grâce à l'aide de la minimisation de la fonction coût.



TAB. 6.23 – Illustration de l'efficacité de l'estimation de la topographie équivalente pour le cas test de la route avec un jeu d'observations partout sauf autour de la route. a- La différence entre la topographie initiale et celle obtenu grâce à la minimisation de la fonction coût. b- Evolution de la valeur de la fonction coût (6.9) en fonction des itérations du code d'optimisation.



TAB. 6.24 – Illustration de l'estimation de la topographie équivalente pour le cas test de la route avec un jeu d'observations uniquement disponible au dessus de la route. a- La topographie du bassin. Ce relief est celui que l'assimilation de données variationnelle doit nous aider à retrouver. b- Le relief permettant de démarrer l'exécution du code d'optimisation, celui-ci est assez éloigné de celui que nous devons retrouver. c- La topographie obtenue grâce à l'aide de la minimisation de la fonction coût.



TAB. 6.25 – Illustration de l'efficacité de l'estimation de la topographie équivalente pour le cas test de la route avec un jeu d'observations uniquement disponible au dessus de la route. a- La différence entre la topographie initiale et celle obtenu grâce à la minimisation de la fonction coût. b- Evolution de la valeur de la fonction coût (6.9) en fonction des itérations du code d'optimisation.

6.2.3 Loire

Soit un domaine carré (50 m × 50 m) bordé à l'ouest par une digue. Au milieu de la digue se trouve une brèche par laquelle l'eau s'infiltre dans le bassin. La condition aux limites ouest sera donc définie grâce à des couples (hauteur, temps).

Les conditions aux limites nord et sud sont de type glissement sans frottement :

$$\vec{u} \cdot \vec{n} = 0$$

et la condition aux limites est de type frontière ouverte

$$\frac{\partial \vec{u}}{\partial \vec{n}} = 0$$

Le temps de l'écoulement sera de 1000 secondes avec un pas de temps de 2 secondes. Le domaine considéré est situé dans le val d'Ouzouer dans la plaine inondable de la Loire. Le relief de cette zone est tiré d'images photogrammétriques et le jeu de Manning a été déduit d'une image SPOT.

Le but de cet exemple est de tester les méthodes présentées précédemment sur un domaine réel.

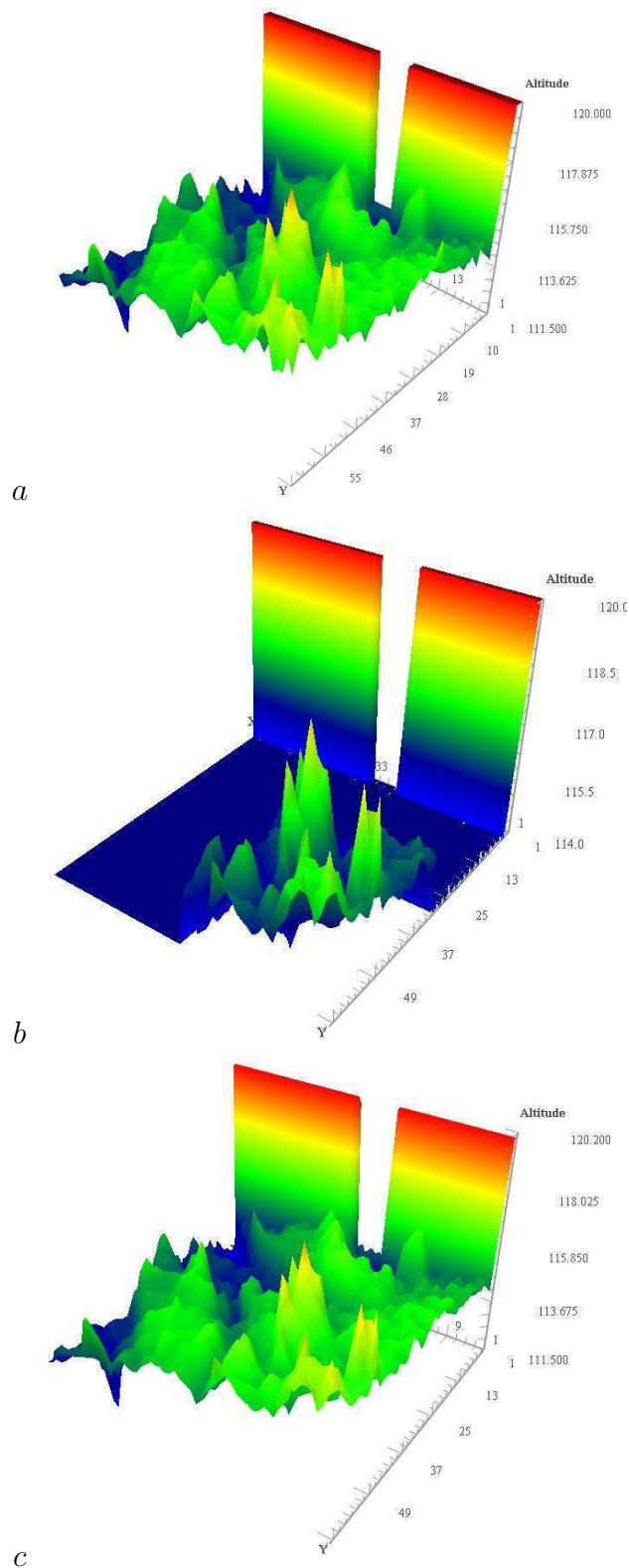
La fonction coût utilisée dans cet exemple est

$$J(Z_b) = \frac{1}{2} \|U - U_{obs}\|^2 + \frac{1}{2} \|V - V_{obs}\|^2 + \frac{1}{2} \|Zn - Zn_{obs}\|^2 \quad (6.10)$$

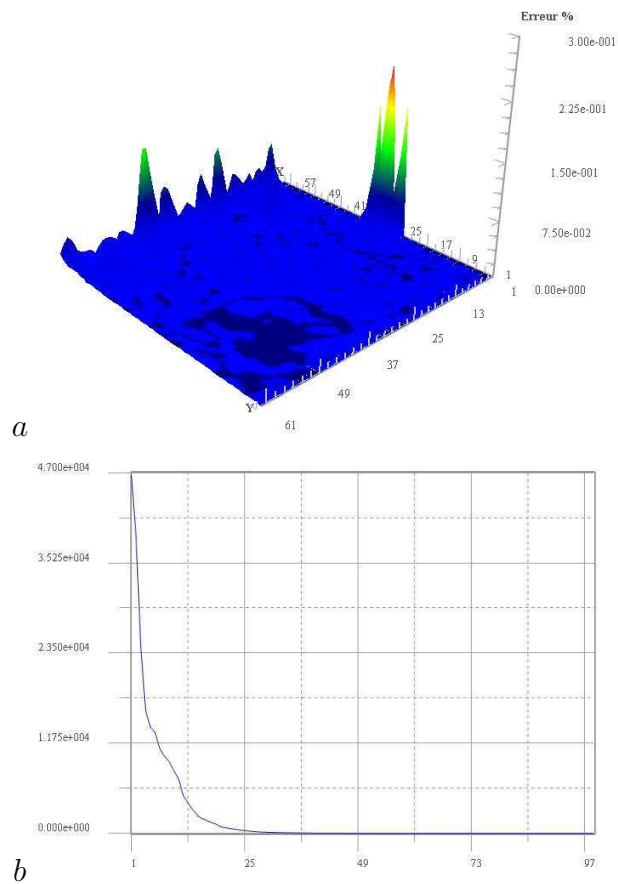
où (U, V) sont les coordonnées de la vitesse, Zn la hauteur d'eau et X_{obs} les observations fournies par la méthode des expériences jumelles (avec $X = \{U, V, Zn\}$). Les observations sont disponibles au temps $t = 1000$.

Afin de mener à bien l'estimation de la topographie équivalente, nous allons initialiser l'exécution du code d'optimisation avec la topographie optimale dont nous avons effacé les trois quart du relief (voir la figure b dans le tableau 6.26).

Malgré la topographie utilisée pour initialiser l'exécution du code d'optimisation, qui est très éloignée de celle à trouver, nous avons pu retrouver une topographie équivalente très proche de celle attendue (voir dans la figure a dans le tableau 6.27 et la figure c dans le tableau 6.26) en quelques itérations du code de minimisation (voir figure b dans le tableau 6.27).



TAB. 6.26 – Illustration de l'estimation de la topographie équivalente pour le cas test de la Loire. a- La topographie du bassin. Ce relief est celui que l'assimilation de données variationnelle doit nous aider à retrouver. b- Le relief permettant de démarrer l'exécution du code d'optimisation, celui-ci est assez éloigné de celui que nous devons retrouver. c- La topographie obtenue grâce à l'aide de la minimisation de la fonction coût.



TAB. 6.27 – Illustration de l'efficacité de l'estimation de la topographie équivalente pour le cas test de la Loire. a- La différence entre la topographie initiale et celle obtenu grâce à la minimisation de la fonction coût. b- Evolution de la valeur de la fonction coût (6.10) en fonction des itérations du code d'optimisation.

6.3 Conclusion

Dans un premier temps, le code adjoint nous a permis de travailler sur l'estimation de la rugosité. Nous avons vu qu'un jeu d'observations dense permet d'optimiser l'estimation de ce paramètre lors de la minimisation de la fonction coût. Que ce soit pour un jeu de Manning uniforme ou très hétérogène, la technique d'assimilation de données variationnelle permet d'obtenir des résultats assez précis. Toutefois, la schématisation de la rugosité peut être nécessaire afin d'en faciliter l'utilisation. Suivant le critère de schématisation choisi, les exemples précédents montrent que l'on peut obtenir un jeu de Manning simplifié qui évite de modifier la caractéristique de l'écoulement

Puis nous nous sommes intéressés à l'estimation d'un paramètre beaucoup plus capital dans la détermination du comportement du flux : la topographie. Grâce à un jeu d'observations complet, l'estimation de ce paramètre peut être très précise. Cependant la dégradation de cette donnée peut dégrader fortement les résultats. En effet, dans l'exemple de la route, nous avons mis en évidence l'importance de la localisation des observations. Ceci pose une question à laquelle il faudra répondre : où doit-on placer les capteurs dans une rivière pour que l'estimation de tel ou tel paramètre soit satisfaisant ?

Un nouveau paramètre peut alors rentrer en compte dans la détermination du réseau de mesure : Le coût de ces installations. En effet, suivant que l'on ait besoin d'une hauteur d'eau ou d'une vitesse le prix de l'installation ne sera pas le même. Peut-être faut-il alors déterminer un réseau optimal de mesure compte tenu de l'analyse de sensibilité et du coût financier ?

Chapitre 7

Analyse de sensibilité

Nous avons étudié quelques exemples liés à l'estimation de différents paramètres comme la rugosité ou le relief. A chaque fois, de nombreuses questions ont été soulevées :

- Quelle est la part de l'apport de la précision des données observées dans la qualité de la prévision ?
- Quelles sont les données importantes à observer ?
- Quel est l'effet de la schématisation sur la prévision ?

Ces questions, dont on devine les enjeux, ont toutes un point commun, elles soulèvent le problème de la sensibilité des variables d'état par rapport au vecteur de contrôle. Comme nous l'avons vu dans le premier chapitre sur l'assimilation de données, le système adjoint au second ordre nous fournit un outil adéquat pour étudier la sensibilité d'un paramètre.

Dans ce chapitre, nous allons tout d'abord étudier la sensibilité de la vitesse du fluide par rapport à la perturbation de la rugosité. Ensuite nous verrons quelle solution peut apporter l'analyse de sensibilité à notre problème de schématisation (voir l'exemple de la section 6.1.5).

Finalement, une étude de la sensibilité des paramètres d'entrée par rapport au vecteur d'observations sera menée afin d'illustrer l'étude sur l'importance de la localisation du jeu d'observations traité dans la section 6.2.2.

7.1 Sensibilité par rapport à la perturbation des rugosités

7.1.1 Calcul de la sensibilité à la perturbation des rugosités sur la Loire

Le domaine considéré dans cet exemple est exactement celui défini dans la section 6.1.5 sur l'estimation de la rugosité.

Lorsque l'on fait une analyse de sensibilité, il faut définir une fonction réponse qui va nous permettre d'étudier la réponse d'une variable à la perturbation d'un paramètre d'entrée ; dans notre exemple ce paramètre est la rugosité.

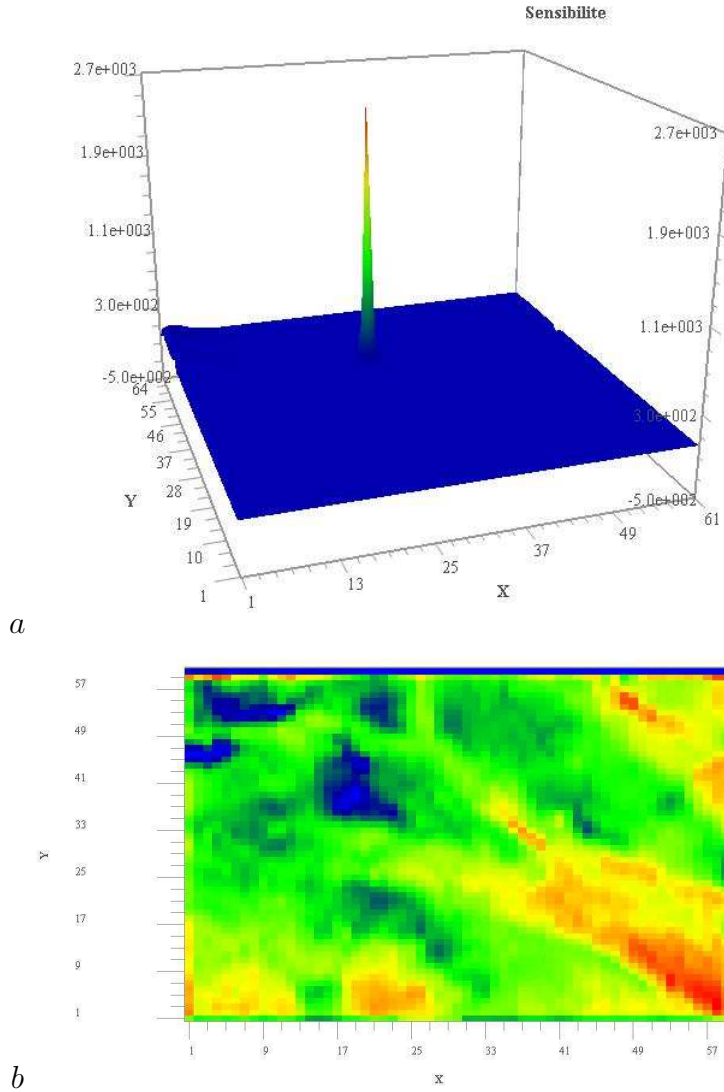
Soit G la fonction réponse :

$$G = u_j^i$$

où u_j^i est la valeur de la prévision de la vitesse U au nœud de la grille $(i\Delta x, j\Delta y)$.

Après la résolution du système adjoint au second ordre et le calcul de la sensibilité comme indiqué dans la section 2.2.2, la sensibilité de la prévision sur la vitesse horizontale par rapport à une perturbation du Manning au point (20,18) est obtenue (cf figure a du tableau 7.1).

Ce graphique montre bien que l'influence de la perturbation du Manning sur la prévision de la vitesse horizontale est localisé sur les points se situant autour du nœud considéré.



TAB. 7.1 – a- Sensibilité de la prévision sur la vitesse U par rapport à la perturbation du coefficient de Manning au point (20,18). b- Carte des sensibilités de la prévision de la vitesse U à la perturbation des coefficients de Manning sur le domaine Loire.

Le calcul de la sensibilité en chaque point de la grille va nous permettre d'obtenir une carte de la sensibilité globale de la prévision de la vitesse U par rapport à la perturbation du coefficient de Manning. Afin de calculer cette carte, nous allons prendre la norme de la sensibilité en chaque point de la grille. Nous obtenons donc une matrice contenant pour chaque point de la grille la norme de la sensibilité correspondante. La figure b du tableau

7.1 est la représentation de cette matrice (ou carte des sensibilités).

Remarque : Sur ce domaine du Val d'Ouzouer, nous pouvons remarquer que la sensibilité est grande à l'endroit où la profondeur est la plus faible (voir figure 7.1). Ceci concorde bien avec les analyses des hydrauliciens.

7.1.2 Utilisation de la sensibilité pour la schématisation de la rugosité

Dans la section 6.1.5, nous avons montré que l'assimilation de données nous permettait de schématiser le jeu de Manning afin d'en faciliter l'usage.

Après avoir considéré plusieurs critères de découpage du domaine en zones de Manning équivalent, nous avons montré que les jeux de Manning obtenus, bien qu'assez éloignés du Manning réel, nous permettaient de calculer un écoulement proche de celui décrit par les observations.

Les critères considérés alors étaient basés sur des considérations géométriques du relief et non pas sur un raisonnement mathématique. L'analyse de sensibilité et par conséquent les cartes de sensibilités qu'elle fournit nous donnent un critère plus logique. En effet, si nous créons de grande zones de Manning équivalent dans les zones du maillage où la sensibilité est faible et nous créons des zones de Manning équivalent fines dans les zones où la sensibilité est importante, il est logique de penser que le découpage ainsi obtenu nous permettra de conserver un écoulement proche de celui attendu.

Sensibilité de la prévision de la vitesse

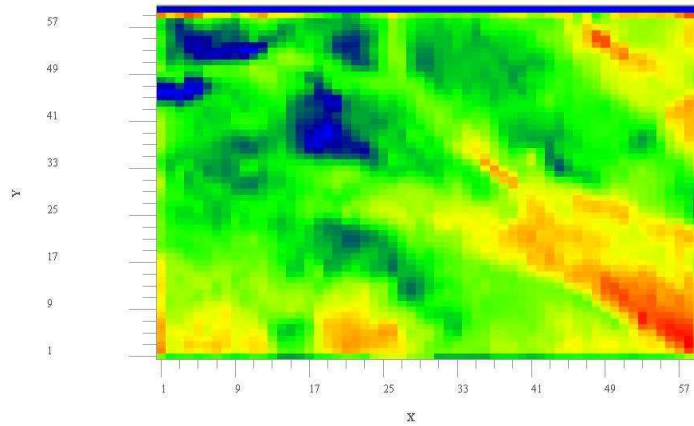


FIG. 7.1 – Carte des sensibilités de la prévision de la vitesse V (en $m^3.s^{-1}$) à la perturbation des coefficients de Manning sur le domaine Loire.

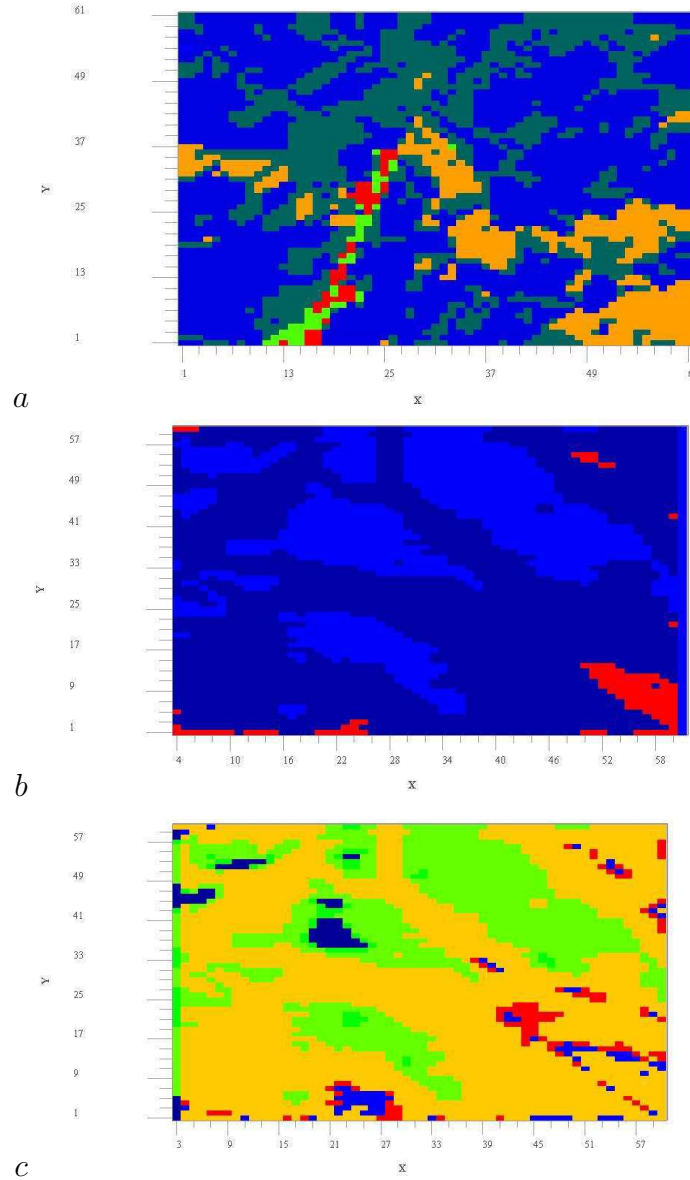
Soit la fonction réponse suivante :

$$G = V_j^i$$

où V_j^i est la valeur de la prévision de la vitesse V au point $(i\Delta x, j\Delta y)$.

Au vu de la figure 7.1, nous pouvons voir que plusieurs zones de sensibilité équivalente apparaissent. Dans un premier temps nous allons découper le domaine en 7 zones de Manning équivalent (voir figure c dans le tableau 7.2) extraites de la carte des sensibilités.

Afin d'appliquer la méthode de l'assimilation de données explicitée dans le premier chapitre, nous allons définir une fonction coût à minimiser.



TAB. 7.2 – *Illustration de l'utilisation de la sensibilité pour la schématisation du jeu de Manning en 7 zones de Manning équivalent. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est la rugosité réelle. b- La rugosité permettant de démarrer l'exécution du code d'optimisation. c- La rugosité obtenue grâce de la minimisation de la fonction coût.*

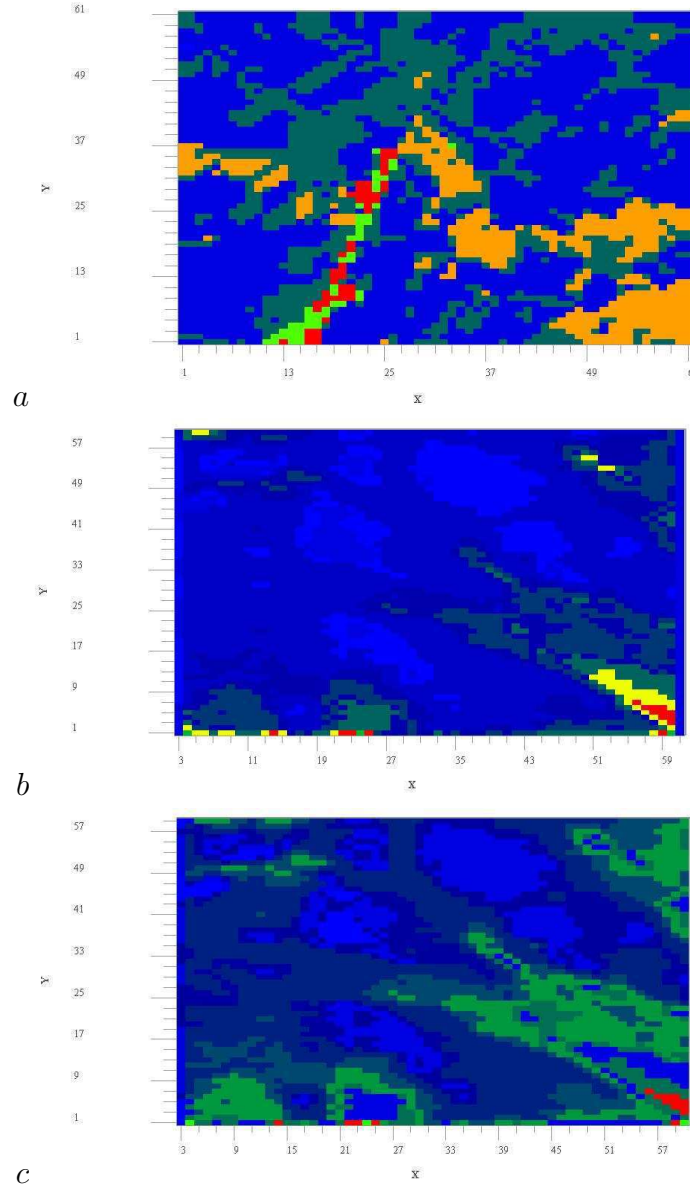
Supposons que nous possédons un jeu d'observations complet, nous pouvons définir J comme suit :

$$J(\text{Manning}) = \frac{1}{2} \|U - U_{obs}\|^2 + \frac{1}{2} \|V - V_{obs}\|^2 + \frac{1}{2} \|Zn - Zn_{obs}\|^2 \quad (7.1)$$

A partir d'un jeu de Manning uniforme, initialisant l'exécution du code d'optimisation, la minimisation de la fonction coût nous fournit un coefficient de rugosité sur chacune des 7

zones. Le minimum de la fonction coût (7.1) étant atteint à 50, nous pouvons en déduire que l'écoulement dans le Val d'Ouzouer est proche de celui attendu.

Un découpage plus fin du domaine, nous fournit à peu près la même précision. Dans le tableau 7.3 se trouvent les résultats pour un découpage en 19 zones de Manning équivalent.



TAB. 7.3 – Illustration de l'utilisation de la sensibilité pour la schématisation du jeu de Manning en 19 zones de Manning équivalent. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est la rugosité réelle. b- La rugosité permettant de démarrer l'exécution du code d'optimisation. c- La rugosité obtenue grâce de la minimisation de la fonction coût.

Sensibilité de la prévision des hauteurs d'eau

Nous allons maintenant mener le même type d'étude que dans la section précédente mais pour la sensibilité des prévisions des hauteurs d'eau par rapport à la perturbation des coefficients de Manning.

Soit la fonction réponse suivante :

$$G = Z_j^i$$

où Z_j^i est la valeur de la prévision de la hauteur d'eau au point $(i\Delta x, j\Delta y)$.

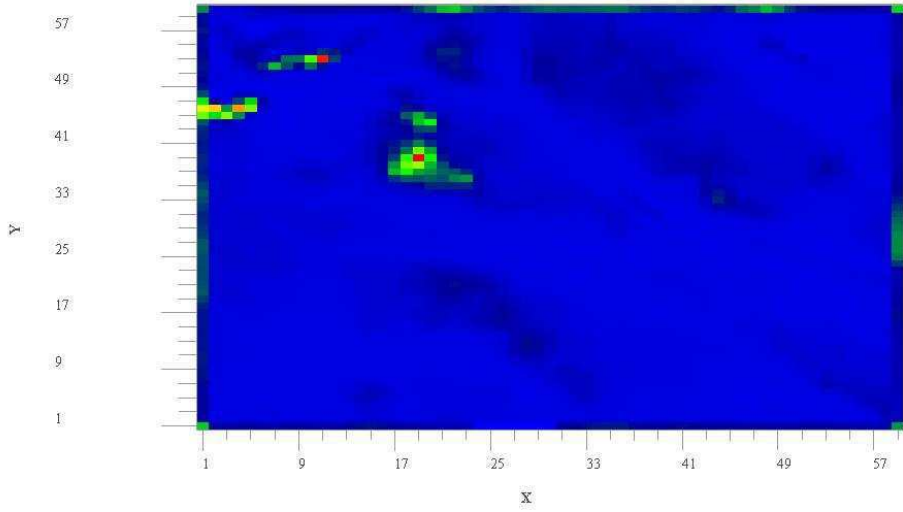
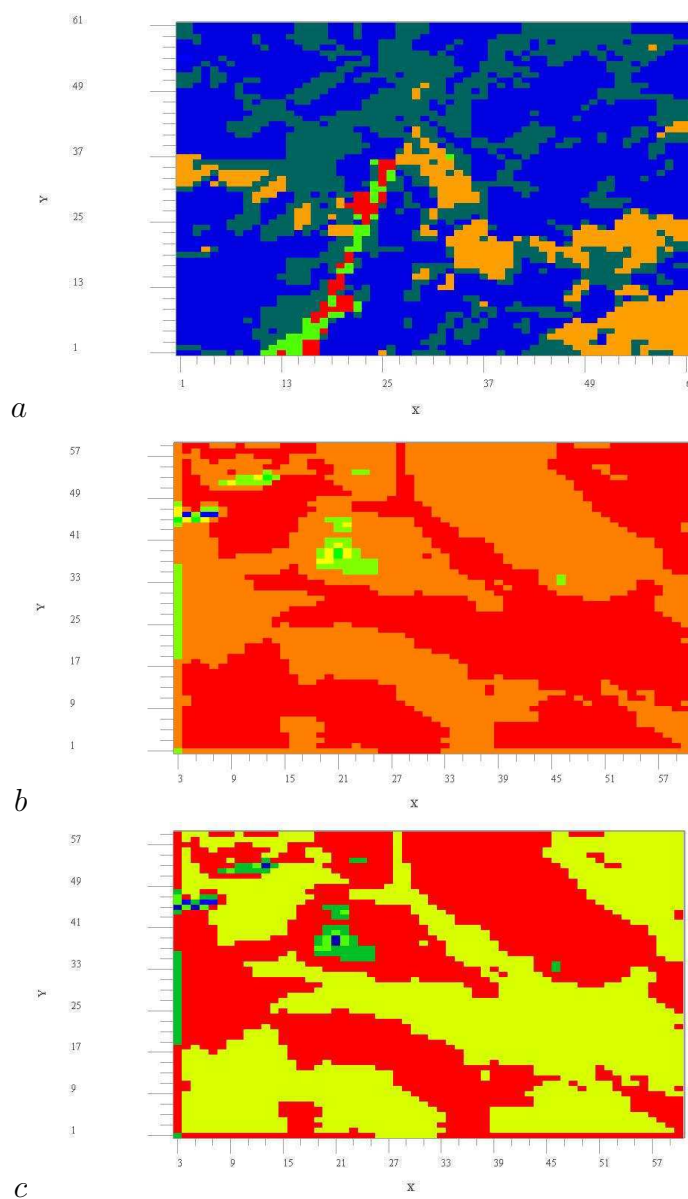


FIG. 7.2 – Carte des sensibilités de la prévision des hauteurs d'eau par rapport à la perturbation des coefficients de Manning sur le domaine Loire.

Comme précédemment, nous découpons le domaine en 7 zones de Manning équivalent (voir le tableau 7.4), mais cette fois-ci le critère de découpage est basé sur la sensibilité de la prévision des hauteurs d'eau donnée par la carte des sensibilités représenté dans la figure 7.2.

La minimisation de la fonction coût (7.1) fournit un coefficient de Manning par zone. Ce minimum est atteint à 60. Le jeu de Manning obtenu (voir la figure c du tableau 7.4) nous permet donc de calculer un écoulement proche de celui observé.



TAB. 7.4 – Illustration de l'utilisation de la sensibilité pour la schématisation du jeu de Manning en 7 zones de Manning équivalent. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est la rugosité réelle. b- La rugosité permettant de démarrer l'exécution du code d'optimisation. c- La rugosité obtenue grâce de la minimisation de la fonction coût.

Le découpage du domaine en 23 zones en fonction de la sensibilité de la prévision des hauteurs d'eau nous donne un résultat équivalent (voir le tableau 7.5). Le découpage en 7 zones est donc suffisant.

7.2 Sensibilité par rapport aux observations

7.2.1 Un exemple trivial

Dans cet exemple nous allons étudier la sensibilité de la prévision des vitesses horizontales par rapport à la perturbation des observations.

Ainsi considérons un bassin à fond plat et à Manning uniforme (cet exemple est celui développé dans la section 6.1.1).

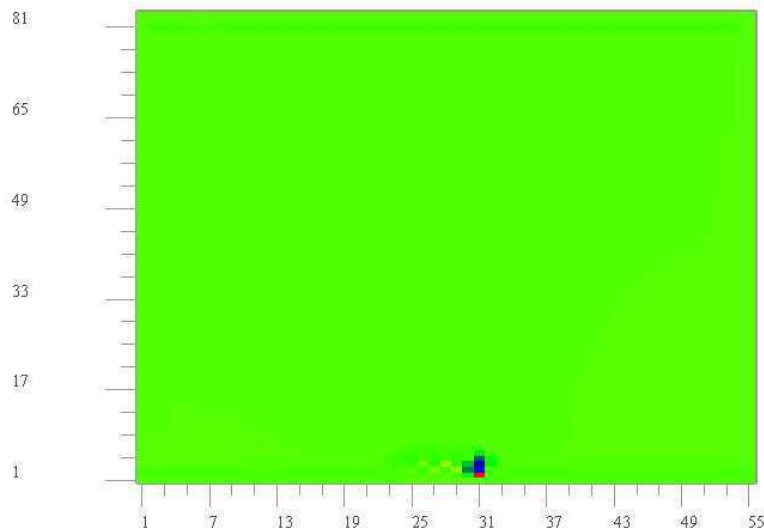


FIG. 7.3 – Carte de la sensibilité de la prévision des vitesses horizontales par rapport aux observations sur un domaine simple à fond plat. Vue de dessus.

Mise à part une petite dégradation des résultats à l'entrée du bassin, la sensibilité est identique sur toute la zone (voir le figure 7.3). Ce résultat est évidemment logique puisque le relief du bassin est uniforme ainsi que l'ensemble des données le définissant. Ainsi, l'influence des observations est identique sur tout le domaine.

7.2.2 Une illustration de la nécessité de bien placer les capteurs

Dans la section 6.2.2, nous avons vu que suivant la localisation des observations, l'estimation du relief équivalent donne des résultats plus ou moins précis.

Dans cet exemple, on considérerait un bassin à fond plat au centre duquel se trouve une route. Afin d'estimer au mieux la géométrie exacte de cette route, nous avons vu qu'il est nécessaire d'avoir des observations au dessus de cette route et si possible autour de celui-ci.

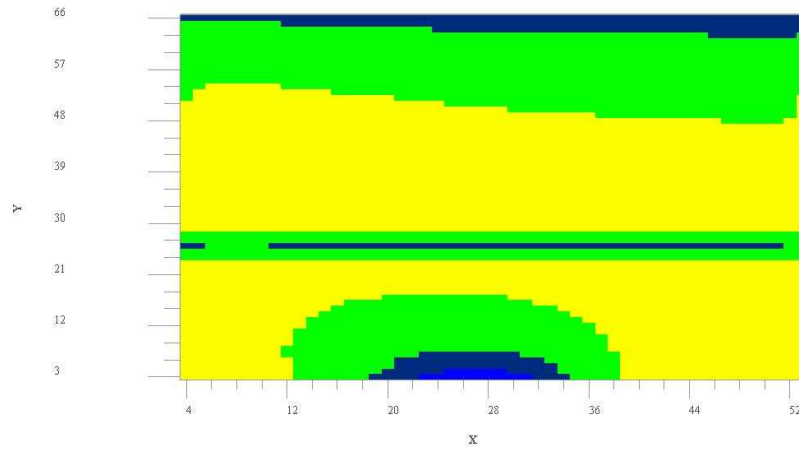


FIG. 7.4 – Carte de la sensibilité de la prévision des hauteurs d'eau par rapport aux observations sur un domaine à fond plat au centre duquel se trouve une route.

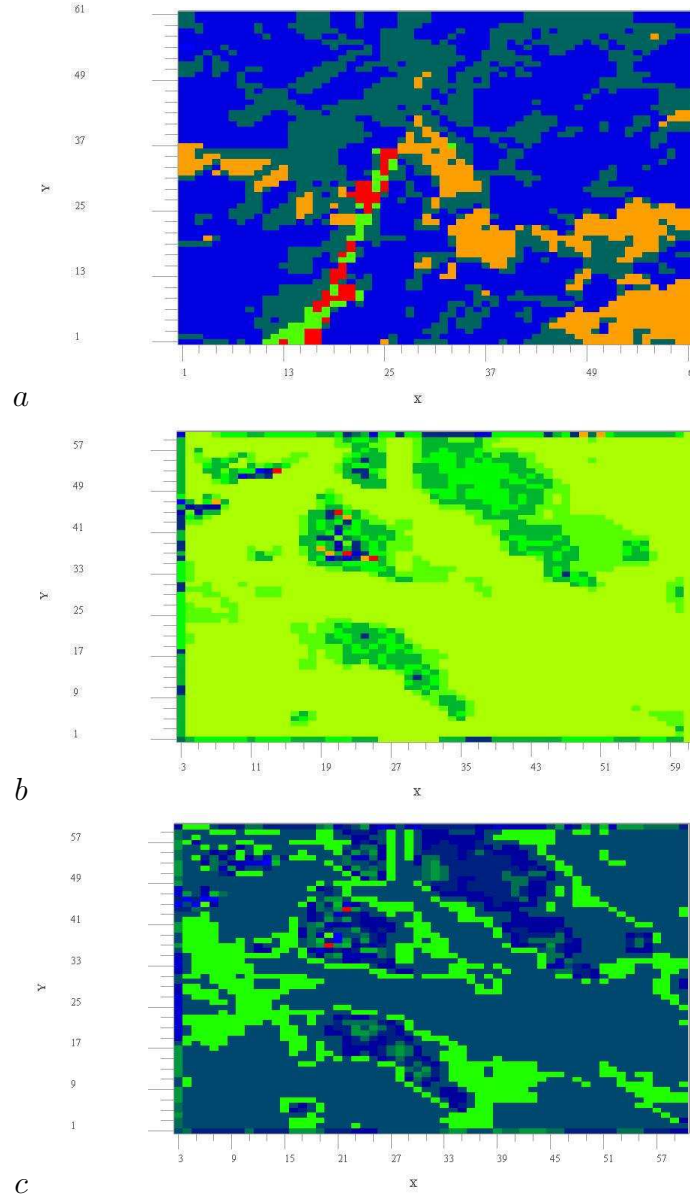
La figure 7.4 qui représente la carte des sensibilités, montre clairement que l'endroit où la sensibilité est la plus forte se trouve directement au dessus de la route ainsi qu'à proximité. Ce résultat confirme donc que les capteurs doivent se situer à cet endroit.

Remarque : Le calcul de la sensibilité se dégrade avec la proximité des bords du domaine. Ainsi, dans le graphique ci-dessus, on voit bien que la sensibilité aux bords du domaine semble aussi importante qu'au dessus de la route alors qu'expérimentalement il paraît inutile d'avoir une forte densité d'observations à cet endroit.

7.3 Conclusion

Nous avons étudié un exemple de schématisation du jeu de rugosité grâce à la sensibilité de la vitesse (ou de la hauteur d'eau) à la perturbation de ce paramètre. Cette schématisation permet de déterminer automatiquement des zones de Manning équivalent. Bien que le jeu de Manning schématisé n'ait rien à voir avec la réalité du terrain, il assure un calcul des résultats du modèle proche de ceux attendus.

Mais l'étude de la sensibilité ne se borne pas à schématiser un jeu de paramètre. Dans la deuxième partie de ce chapitre, nous avons répondu au problème posé par la localisation optimale des observations. C'est à dire, que nous avons déterminé quel est le jeu d'observations minimal nécessaire à une bonne étude pour un domaine donné. L'analyse de sensibilité permet ainsi de déterminer de manière précise quelles sont les observations les plus importantes pour la détermination des variables de contrôle du modèle utilisé.



TAB. 7.5 – Illustration de l'utilisation de la sensibilité pour la schématisation du jeu de Manning en 23 zones de Manning équivalent. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est la rugosité réelle. b- La rugosité permettant de démarrer l'exécution du code d'optimisation. c- La rugosité obtenue grâce de la minimisation de la fonction coût.

Chapitre 8

Assimilation de données et parallélisation

Modéliser l'écoulement d'un bassin versant nécessite de choisir un modèle d'hydraulique fluviale en fonction du but de l'étude. Le comportement de l'écoulement dans un grand bassin versant se modélisera plutôt grâce à un modèle monodimensionnel afin de ne pas dépasser les capacités mémoire de la machine. A contrario, pour étudier le comportement de l'eau près d'une pile de pont, il faudra choisir de préférence un modèle tridimensionnel.

Mais le but de l'étude peut nécessiter plusieurs degrés de précision. C'est à dire que l'on peut vouloir étudier l'écoulement global d'une rivière dans un grand bassin versant et l'écoulement précis d'une petite partie de ce domaine. Afin de résoudre ce problème, il faut alors coupler deux types de modèle ; une modèle monodimensionnel afin de résoudre les équations de l'hydraulique fluviale sur le domaine du grand bassin versant et un modèle bidimensionnel pour calculer le comportement de l'écoulement dans le petit domaine qui requiert une précision supérieure.

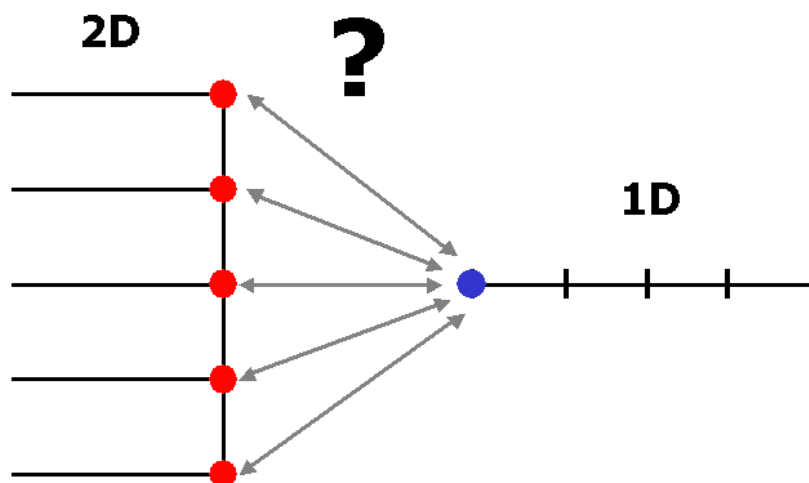


FIG. 8.1 – *Problème posé par le couplage d'un modèle monodimensionnel avec un modèle bidimensionnel.*

Cependant, ce couplage soulève un problème illustré par la figure 8.1. Quelles valeurs utiliser aux interfaces entre les modèles ? C'est-à-dire lorsque l'on couple un modèle 2D avec un modèle 1D, nous avons à l'interface plusieurs valeurs de condition aux limites provenant du modèle 2D et une valeur venant du modèle 1D. Si l'écoulement se propage du modèle 2D vers le modèle 1D, quelle valeur doit-on imposer en amont du modèle 1D, sachant que celle-ci doit être calculée grâce aux valeurs provenant de l'aval du modèle 2D ? Généralement, une interpolation linéaire est utilisée afin de trouver une valeur à communiquer. Mais le choix de cette interpolation est discutable puisqu'il ne dépend pas des observations.

Nous allons voir dans ce chapitre, l'utilisation de la méthode de l'assimilation de données afin de contrôler les interfaces entre les modèles tout en estimant une condition initiale pour chaque domaine.

8.1 Décomposition de domaine

Afin d'illustrer la problème du couplage de modèles nous allons considérer une rivière Ω subdivisée en plusieurs morceaux. L'écoulement dans chaque sous domaine sera calculé par le même modèle. L'assimilation de donnée sera utilisée afin de contrôler les interfaces entre les sous domaines.

Soit une rivière Ω dont l'écoulement est calculé par le modèle discrétisé suivant :

$$\begin{aligned} \frac{dX}{dt} + F(X) + B.CL &= 0 \\ X(0) &= U \end{aligned} \quad (8.1)$$

où F est la discrétisation d'un opérateur d'équations aux dérivées partielles, X est la variable d'état du système, CL la condition aux limites et U la condition initiale. U et CL seront les paramètres de contrôle de notre étude. Ce système sera résolu sur l'intervalle de temps $[0, T]$.

Soit X_{obs} , un jeu de données que nous considérerons continu en temps, χ l'espace des prévisions tel que $X \in \chi$ et χ_{obs} celui des observations tel que $X_{obs} \in \chi_{obs}$. Soit l'opérateur C projetant une variable de l'espace des prévisions sur celui des observations :

$$C : \chi \rightarrow \chi_{obs}$$

Comparons maintenant mesures et observations grâce à la fonctionnelle J que nous cherchons à minimiser :

$$J(U, CL) = \frac{1}{2} \|CX(U, CL) - X_{obs}\|_{\chi_{obs}}^2 + \frac{1}{2} \|U - U_0\|_{\mathcal{U}}^2 + \frac{1}{2} \|CL - CL_0\|_{\mathcal{CL}}^2 \quad (8.2)$$

avec $\|\cdot\|_{\chi_{obs}}^2$ la norme définie dans l'espace des observations, $\|\cdot\|_{\mathcal{U}}^2$ celle de l'espace dans lequel se trouve les conditions initiales possibles et $\|\cdot\|_{\mathcal{CL}}^2$ celle de l'espace dans lequel se trouve les conditions aux limites possibles. U_0 est la valeur *a priori* de la condition initiale et CL_0 celle de la condition aux limites.

Le problème est de trouver U^* et CL^* minimisant J et caractérisées par la condition d'Euler :

$$\begin{aligned} \nabla_U J(U^*) &= 0 \\ \nabla_{CL} J(CL^*) &= 0 \end{aligned} \quad (8.3)$$

Après de nombreux calculs détaillés dans les sections 2.1.1 et 2.1.3, nous obtenons la caractérisation suivante du minimum de J en fonction de la variable adjointe p :

$$\begin{cases} \nabla_U J &= -p(0) \\ \nabla_{CL} J &= B^t \cdot p \end{cases} \quad (8.4)$$

Soit la rivière Ω divisé en n sous domaines Ω_i (cf figure 8.2).

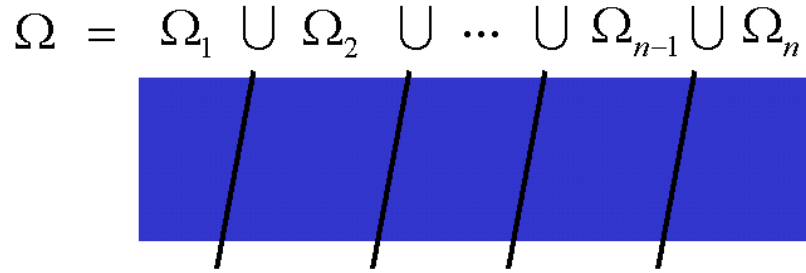


FIG. 8.2 – *Subdivision de la rivière Ω en n sous domaines.*

Considérons que l'évolution de l'écoulement dans chaque sous domaine est résolu sur un processeur indépendant. Le but de cette étude est d'optimiser la condition aux limites à l'interface entre deux sous domaines et la condition initiale sur chaque sous domaine grâce à la méthode de l'adjoint introduite précédemment. Cette approche s'apparente à une parallélisation non cohérente (au sens du découpage de domaine) des méthodes d'assimilation de données.

Posons X la variable d'état sur le domaine entier :

$$\begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_i \\ \vdots \\ X_n \end{pmatrix} \begin{array}{l} \leftarrow \text{Variable d'état sur le domaine } \Omega_1 \\ \leftarrow \text{Variable d'état sur le domaine } \Omega_2 \\ \vdots \\ \leftarrow \text{Variable d'état sur le domaine } \Omega_i \\ \vdots \\ \leftarrow \text{Variable d'état sur le domaine } \Omega_n \end{array}$$

Le problème maintenant est de résoudre le système (8.1) sur chaque sous domaine, c'est-à-dire sur chaque processeur, ainsi pour le sous domaine Ω_i :

$$\begin{aligned} \frac{dX_i}{dt} + F(X_i) + B_1 \cdot CL_i &= 0 \\ X_i(0) &= U_i \end{aligned} \quad (8.5)$$

A chaque interface entre deux sous domaines, on trouve 2 valeurs de CL , une provenant de l'amont et l'autre de l'aval. Si les deux valeurs sont égales, on a alors la bonne solution sur tout le domaine.

Pour estimer les conditions initiale et aux limites nous devons minimiser une fonction coût. Afin d'assurer la continuité de la condition initiale entre les sous domaines nous devons pénaliser la fonction coût par la mesure de la différence des deux valeurs aux interfaces du domaine.

Par conséquent, introduisons la fonction coût qui devra être minimisée sur le sous domaine Ω_i :

$$J(U_i, CL_i) = \frac{1}{2} \|C_i \cdot X_i - X_{i,obs}\|_{\chi_{obs}}^2 + \frac{1}{2} \|CL_{1,i} - CL_{2,i-1}\|_{\mathcal{CL}}^2 + \frac{1}{2} \|CL_{2,i} - CL_{1,i+1}\|_{\mathcal{CL}}^2 \quad (8.6)$$

avec $CL_{1,i}$ la condition aux limites amont du sous domaine Ω_i et $CL_{2,i}$ la condition aux limites aval du sous domaine Ω_i .

De plus et pour assurer une meilleure continuité de la condition initiale nous allons introduire dans la minimisation de la fonction coût, une méthode itérative qui va donner de plus en plus de poids aux termes de pénalisation. Ainsi, pour chaque sous domaine Ω_i et à l'itération k , nous allons minimiser la fonction coût suivante :

$$J(U_i^k, CL_i^k) = \frac{1}{2} \|C_i \cdot X_i^k - X_{i,obs}^k\|_{\chi_{obs}}^2 + \frac{1}{2-k} \|CL_{1,i}^k - CL_{2,i-1}^{k-1}\|_{\mathcal{CL}}^2 + \frac{1}{2-k} \|CL_{2,i}^k - CL_{1,i+1}^{k-1}\|_{\mathcal{CL}}^2 \quad (8.7)$$

Remarque : Si C est diagonal par bloc cela signifie que les sous domaines sont indépendants. Les problèmes de contrôle optimal sont alors indépendants et peuvent être traités sur des processeurs différents, la communication entre ces derniers se limitera alors à l'échange des valeurs aux bords du sous domaine entre chaque itération.

8.2 Mise en œuvre de la méthode

8.2.1 Modèle général

Afin de tester la méthode exposée précédemment, nous allons définir une rivière où l'écoulement sera calculé par le logiciel LMCFLD.

Soit un domaine carré ($50m \times 50m$) bordé au nord et au sud par des digues infranchissable. Les conditions aux limites nord et sud sont de type glissement sans frottement :

$$\vec{u} \cdot \vec{n} = 0$$

La condition aux limites imposée en amont est celle présentée dans la figure 8.3.

Le temps de l'écoulement sera de 100 secondes avec un pas de temps de 2 secondes. Le jeu de Manning, représentant la nature du sol, est uniformément égal à 0.05. La pente du lit de la rivière est de l'ordre de 2%.

Le but de cette étude sera de découper la rivière en n domaines, de calculer l'écoulement sur chaque sous domaine à l'aide du logiciel LMCFLD et d'estimer conjointement les conditions initiale et aux limites sur chaque sous domaine. Le calcul de ces écoulements sera pris en charge par des processeurs différents. Dans la section suivante nous allons présenter rapidement le cluster utilisé.

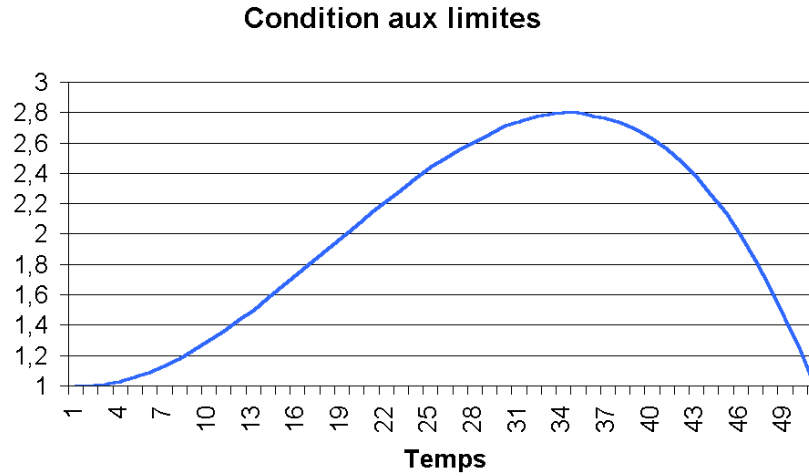


FIG. 8.3 – Condition aux limites imposée en amont dans le cas test illustrant la méthode de décomposition de domaine.

8.2.2 Hardware

MIRAGE est une des composantes du projet CIMENT, qui vise à promouvoir et à coordonner la méso-informatique au sein de l'Université Joseph Fourier autour de thématiques scientifiques proches (Environnement et Climat) ayant en commun l'utilisation de modélisations numériques nécessitant des ressources de calcul importantes (voir site Web <http://mirage.imag.fr/>).

Les machines du projet MIRAGE sont réparties sur 3 sites du Campus Universitaire de Saint Martin d'Hères

- LMC : 2 DEC¹ DS20 + 1 baie disque RA3000
- LEGI/LTHE : 4 DEC DS20 + 1 baie disque RA3000
- LGGE : 1 DEC DS20 + 1 baie disque RA3000

La plate forme MIRAGE est utilisée pour du calcul distribué au travers le réseau du campus de l'UJF. Actuellement le LEGI/LTHE et le LMC sont reliés par un réseau à 1Gigabit, tandis que la liaison vers les autres laboratoires est à 100Mb.

Pour éviter que le speed up des exemples que nous présenterons dans les sections suivantes dépendent de la différence de performance entre les liens qui lient les différents réseaux, nous nous contenterons d'utiliser les machines du LMC et du LEGI/LTHE.

8.2.3 Parallélisation

L'écoulement dans chaque sous domaine est donc calculé par un processeur indépendant (la décomposition de domaine a été effectuée sans recouvrement). La communication entre les sous domaines (i.e. entre les processeurs) se limitera à l'envoi et à la réception des valeurs aux limites entre sous domaines voisins.

Sur chaque sous domaine la fonction coût (8.7) sera minimisée afin d'estimer la condition initiale et les conditions aux limites optimales sachant que les observations seront dans un premier temps disponibles à chaque pas de temps et pour chaque point de la grille.

¹bi-processeurs : 500MHz/EV6, 2 disques internes : 2 x 18 Gb (dont espace de travail + Swap), mémoire : 2.5 Go / machine, Mémoire Cache : 4 Mo / machine

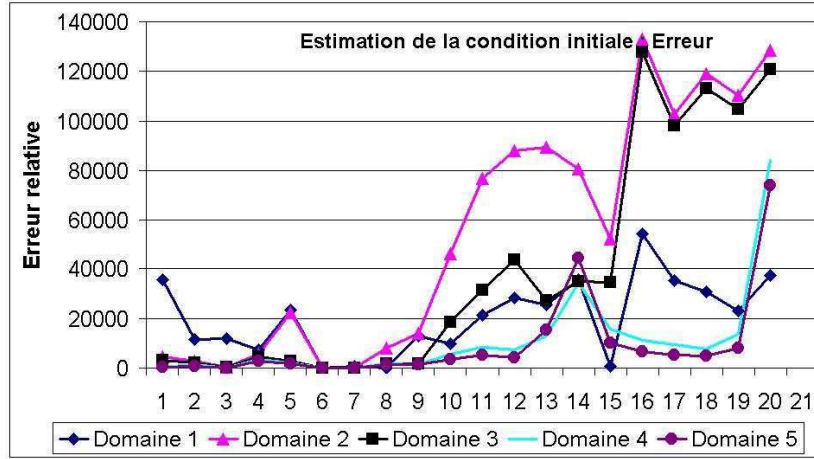


FIG. 8.4 – Normes de l'erreurs relatives commises sur les 5 sous domaines et à chaque itération k afin de retrouver la condition initiale pour le cas test de la rivière avec un jeu d'observations complet. La norme utilisée ici est la norme de la matrice des erreurs relatives commises sur le domaine (les éléments de la matrice répertorient les erreurs relatives commises en chaque point de la grille).

8.3 Une décomposition régulière

Le rivière Ω présentée précédemment est subdivisée en 5 sous domaines de tailles identiques dans la direction de l'écoulement (ouest-est). La taille de la rivière Ω est $50m \times 50m$ chaque sous domaine sera donc de taille $10m \times 50m$.

Afin d'initialiser l'exécution du code d'optimisation, nous allons perturber aléatoirement la condition initiale à retrouver sur chaque sous domaine. De plus la condition aux limites amont utilisée sera constante dans le temps.

Un jeu d'observation est disponible grâce à l'utilisation de la méthode des expériences jumelles.

8.3.1 Un exemple avec un jeu d'observation complet

Le jeu d'observation nécessaire à la définition de la fonction coût (8.7) est complet. C'est-à-dire que nous avons une observation à chaque point de la grille et pour chaque pas de temps.

Tout d'abord nous allons regarder les résultats sur l'estimation de la condition initiale. La figure 8.4 montre l'évolution de l'erreur commise dans l'estimation de la condition initiale itérations après itérations.

On peut voir que dès la première itération, la condition initiale est bien retrouvée sur chaque sous domaine. A partir de l'itération 10, l'erreur commence à augmenter, le terme de pénalisation devient trop important par rapport au terme mesurant la différence entre les observations et les prévisions. La méthode de recherche des extrema se focalise donc vers la minimisation des termes de pénalisation.

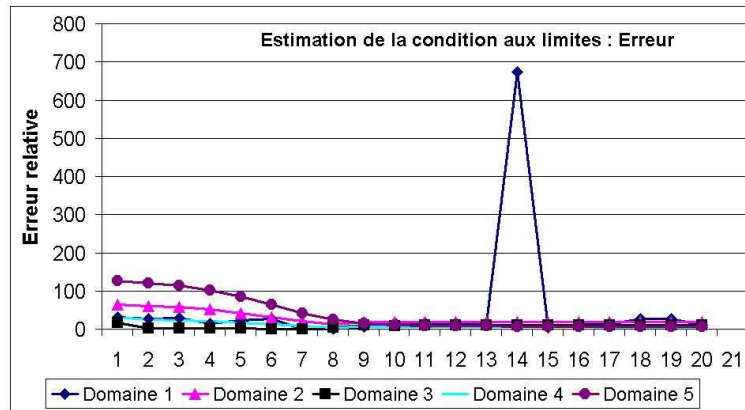


FIG. 8.5 – Normes de l'erreurs relatives commises sur les 5 sous domaines et à chaque itération k afin de retrouver la condition aux limites amont pour le cas test de la rivière avec un jeu d'observations complet.

Voyons maintenant, les résultats sur l'estimation de la condition aux limites amont. La figure 8.5 montre l'évolution de l'erreur commise afin de retrouver la condition aux limites amont en fonction de l'itération k . Cette figure montre clairement que sur tout les domaines l'estimation de la condition aux limites est de mieux en mieux retrouvée itération après itération. A partir de la 10ème itération l'estimation de cette condition semble avoir atteint son optimum.

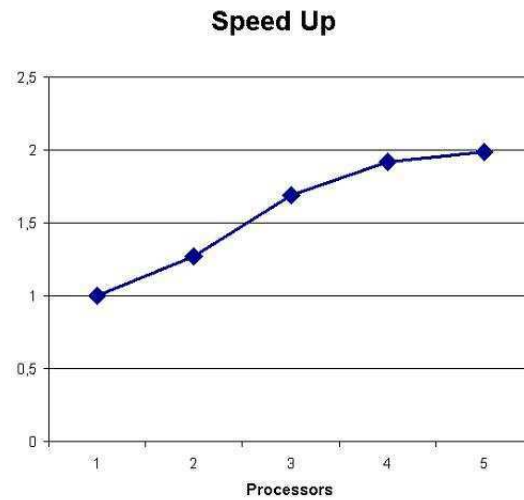


FIG. 8.6 – Speed Up obtenu sur la plateforme Mirage pour le cas test de la rivière avec un jeu d'observations complet.

La figure 8.6 montre le speed up obtenu sur la plateforme Mirage. Avec 5 processeurs nous obtenons un gain de temps de facteur 2 grâce à la méthode de la décomposition de domaine.

Dans les chapitres précédent nous sommes arrivés à la conclusion que le jeu d'observations devait être judicieusement choisis si l'on désirait obtenir une estimation fiable. Nous

allons maintenant dégrader le jeu d'observations et tester notre méthode de découpage de domaine avec ce jeu dégradé.

8.3.2 Un exemple avec un jeu d'observation dégradé

Supposons que le jeu d'observation soit complet sur tout le domaine excepté sur le domaine numéro 3 où aucune observation n'est disponible.

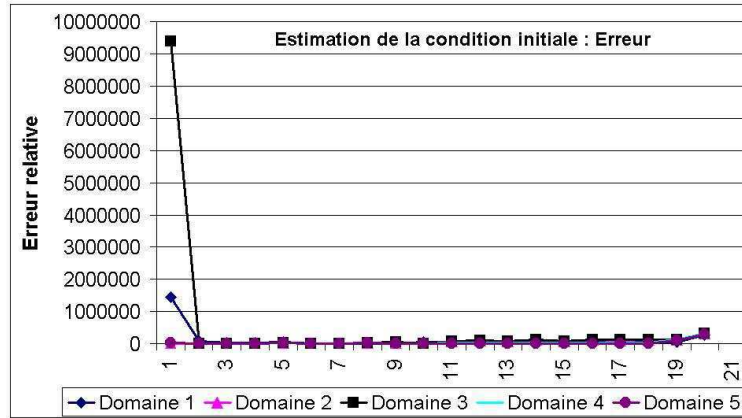


FIG. 8.7 – Normes de l'erreurs relatives commises sur les 5 sous domaines et à chaque itération k afin de retrouver la condition initiale pour le cas test de la rivière avec un jeu d'observations dégradé.

La figure 8.7 expose les erreurs commises afin d'estimer les conditions initiales sur chaque sous domaine. Le manque d'observations sur le sous domaine 3 pénalise forcément la qualité de l'estimation dans les premières itérations. Mais après quelques itérations, on arrive tout de même à estimer la condition sur le sous domaine 3 de manière satisfaisante puisque l'erreur relative moyenne à la troisième itération est de 1%.

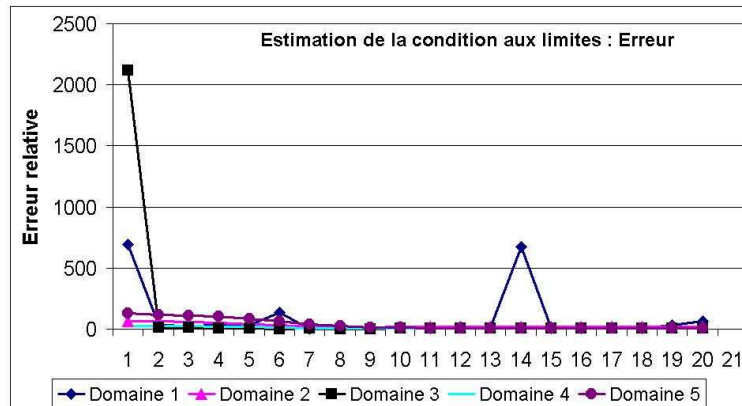


FIG. 8.8 – Normes de l'erreurs relatives commises sur les 5 sous domaines et à chaque itération k afin de retrouver la condition aux limites amont pour le cas test de la rivière avec un jeu d'observations dégradé.

La figure 8.8 montre que la dégradation du jeu d'observation ne pénalise en rien l'estimation de la condition aux limites sur les domaines 1, 2, 4 et 5. Par contre sans observations, l'estimation de la condition aux limites sur le troisième domaine n'est pas aussi bien retrouvé que précédemment. En effet, nous voyons qu'à la première itération l'estimation de cette condition est très mauvaise. Par contre après plusieurs itérations cette estimation s'améliore nettement grâce aux termes de pénalisation qui tiennent compte des conditions aux limites voisines qui elles sont bien retrouvée grâce aux observations disponible sur les sous domaine voisins.

La qualité des résultats obtenus est surprenante puisque le jeu d'observation a été dégradé de manière significative. On peut se demander si la qualité des résultats n'est pas la conséquence de la taille du sous domaine par rapport à ses sous domaines voisins. Dans la section suivante nous allons découper la rivière en sous domaine de taille très différentes.

8.4 Une décomposition irrégulière

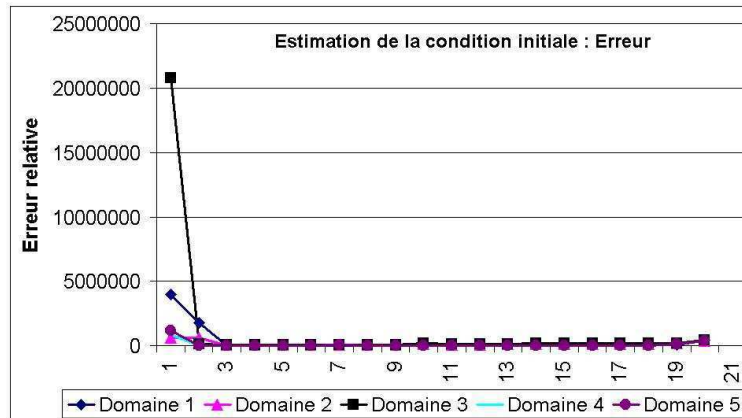


FIG. 8.9 – Normes de l'erreurs relatives commises sur les 5 sous domaines et à chaque itération k afin de retrouver la condition initiale pour le cas test de la rivière avec un jeu d'observations dégradé et un sous domaine numéro 3 de taille disproportionné.

La rivière Ω présentée précédemment est subdivisée en 5 sous domaines de tailles différentes dans la direction de l'écoulement (ouest-est). Les tailles des sous domaines sont exposées dans la tableau suivant :

	Ouest-Est	Nord-Sud
Sous domaine 1	7	50
Sous domaine 2	8	50
Sous domaine 3	20	50
Sous domaine 4	8	50
Sous domaine 5	7	50

Afin d'initialiser l'exécution du code d'optimisation, nous allons perturber aléatoirement la condition initiale à retrouver sur chaque sous domaine. De plus la condition aux limites

amont utilisée sera constante dans le temps.

Un jeu d'observation est disponible grâce à l'utilisation de la méthode des expériences jumelles.

Un jeu d'observation dégradé

Comme dans l'exemple précédent nous avons un jeu d'observations complet sur tout les sous domaines excepté sur le sous domaine numéro 3 où aucune observation n'est disponible.

Ce nouveau découpage pénalise forcément la qualité des résultats. En effet, bien que la figure 8.9 montre que l'on converge vers une condition initiale proche de celle obtenue, l'erreur relative moyenne est de 18%.

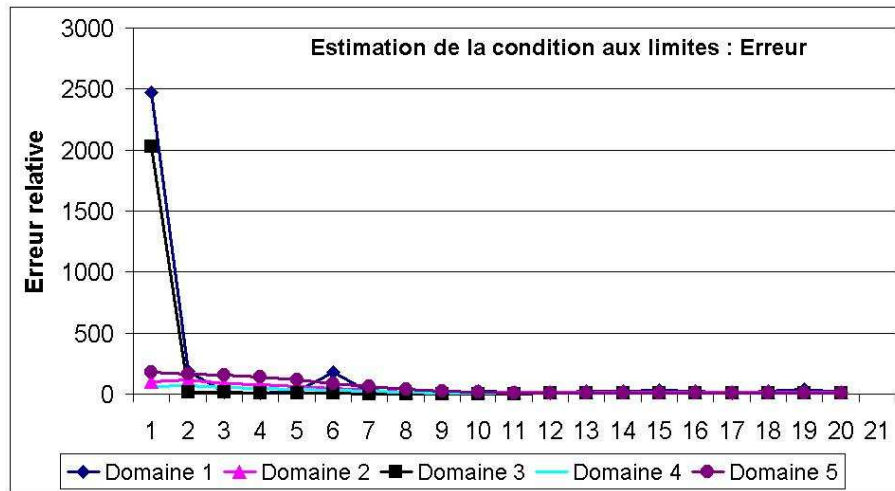
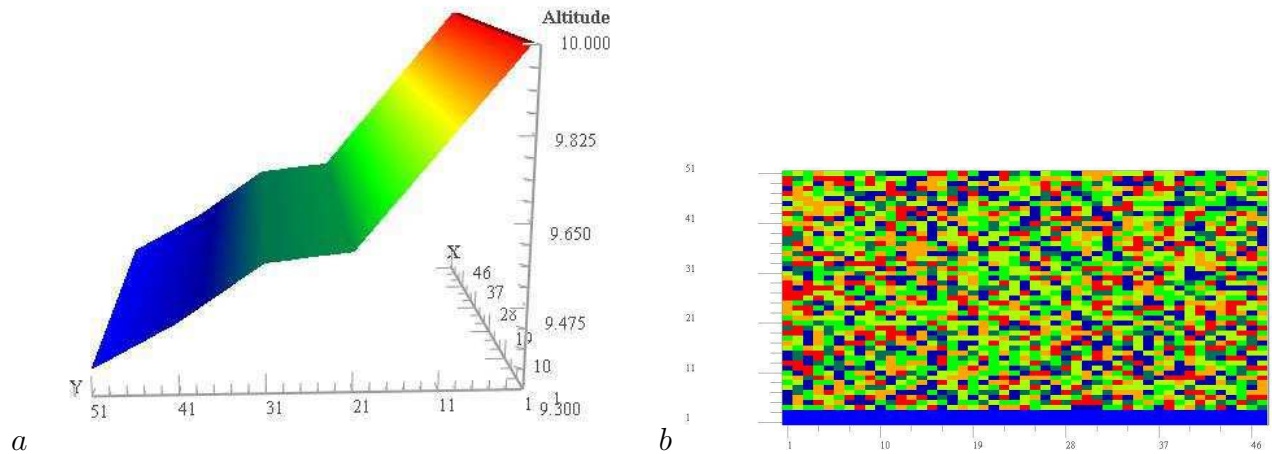


FIG. 8.10 – Normes de l'erreurs relatives commises sur les 5 sous domaines et à chaque itération k afin de retrouver la condition aux limites amont pour le cas test de la rivière avec un jeu d'observations dégradé un sous domaine numéro 3 de taille disproportionné.

Par contre, l'erreur commise afin d'estimer une condition aux limites optimale n'a pas été fortement pénalisée par le changement de taille du domaine 3 (voir figure 8.10).

On peut toutefois être surpris pas la qualité des résultats. Même pour cet exemple où l'erreur passe de 1% à 18%, ce résultat paraît encore satisfaisant. Bien sur plus on va faire grandir la taille du domaine numéro 3 plus l'erreur va augmenter, ce résultat pourrait nous inciter à découper un domaine en fonction de sa densité en observations disponible. Cependant, l'exemple traité ici est trop régulier pour nous inciter à répondre si facilement. En effet, le fond du lit de la rivière est plat et de pente très faible, la rugosité est identique sur tout le domaine, l'eau peut donc s'écouler docilement sans aucune difficultés. Nous devons donc envisager un exemple plus complexe afin de tester si l'estimation des conditions aux limites et initiales sont pénalisée par une géométrie plus perturbée.



TAB. 8.1 – Relief du lit de la rivière (a) et distribution de la rugosité (b) dans le cas test illustrant la méthode de décomposition de domaine avec un exemple plus complexe.

8.5 Un exemple de rivière dont le relief et la rugosité sont plus "chaotique"

Pour créer cet exemple, nous allons reprendre l'exemple précédent et modifier le relief du lit de la rivière et la distribution des coefficients de rugosité.

Nous allons définir un relief composé de trois tronçons de pentes différentes (voir figure a-8.1) puis une répartition de Manning aléatoire (voir figure b-8.1).

Un jeu d'observation dégradé

Comme dans l'exemple précédent nous avons un jeu d'observations complet sur tous les sous domaines excepté sur le sous domaine numéro 3 où aucune observation n'est disponible.

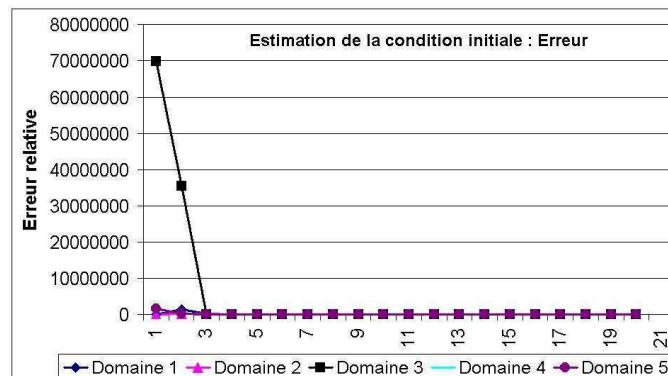


FIG. 8.11 – Normes de l'erreurs relatives commises sur les 5 sous domaines et à chaque itération k afin de retrouver la condition initiale pour le cas test de la rivière "chaotique" avec un jeu d'observations dégradé et un sous domaine numéro 3 de taille disproportionné.

De même que pour une rivière régulière, ce nouveau découpage pénalise la qualité des résultats. En effet, bien que la figure 8.11 montre que l'on converge vers une condition initiale proche de celle obtenue, l'erreur relative moyenne est de 20% au minimum.

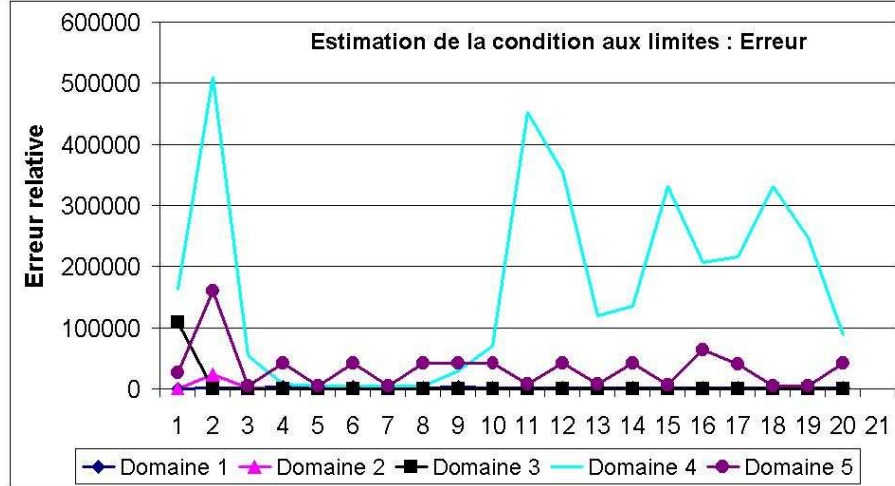


FIG. 8.12 – Normes de l'erreurs relatives commises sur les 5 sous domaines et à chaque itération k afin de retrouver la condition aux limites amont pour le cas test de la rivière "chaotique" avec un jeu d'observations dégradé et un sous domaine numéro 3 de taille disproportionné.

Par contre, l'erreur commise afin d'estimer une condition aux limites optimale n'a pas été fortement pénalisée par le changement de relief et de rugosité (voir figure 8.12).

Par conséquent, pour les exemples étudiés, la dégradation du jeu d'observations n'est pas forcément pénalisante pour une estimation fiable des conditions aux limites et initiales. Cette distribution des observations peu même être un critère de découpage de la rivière.

Nous avons vu que la méthode itérative introduite dans la définition de la fonction coût (8.7) permet d'améliorer nettement l'estimation des variables. Cependant les exemples précédent ont été menés en fixant arbitrairement un nombre maximum d'itérations à ne pas dépasser. Dans la section suivante nous allons envisager la recherche d'un critère d'arrêt à la méthode itérative.

8.6 Vers un critère d'arrêt de la méthode itérative

Afin de mener à bien cette étude du critère d'arrêt de la méthode itérative nous allons reprendre l'exemple de la rivière régulière introduite précédemment. Le découpage de celle-ci sera donc comme suit :

	Ouest-Est	Nord-Sud
Sous domaine 1	10	50
Sous domaine 2	10	50
Sous domaine 3	10	50
Sous domaine 4	10	50
Sous domaine 5	10	50

Le premier critère qui vient à l'esprit est la valeur de la fonction coût. En effet, comme c'est cette fonction qui va être minimisée, il est logique de se dire que lorsque son minimum est atteint il faudra stopper la progression du coefficient k . Cependant, plus k est grand plus les termes de pénalisation vont prendre de l'importance et l'emporter sur le terme $\frac{1}{2} \|C_i \cdot X_i^k - X_{i,obs}^k\|_{\chi_{obs}}^2$.

Nous allons donc choisir comme critère d'arrêt la valeur des termes de pénalisation, c'est-à-dire

$$\begin{cases} \|CL_{1,i}^k - CL_{2,i-1}^{k-1}\|_{\mathcal{CL}}^2 \\ \|CL_{2,i}^k - CL_{1,i+1}^{k-1}\|_{\mathcal{CL}}^2 \end{cases}$$

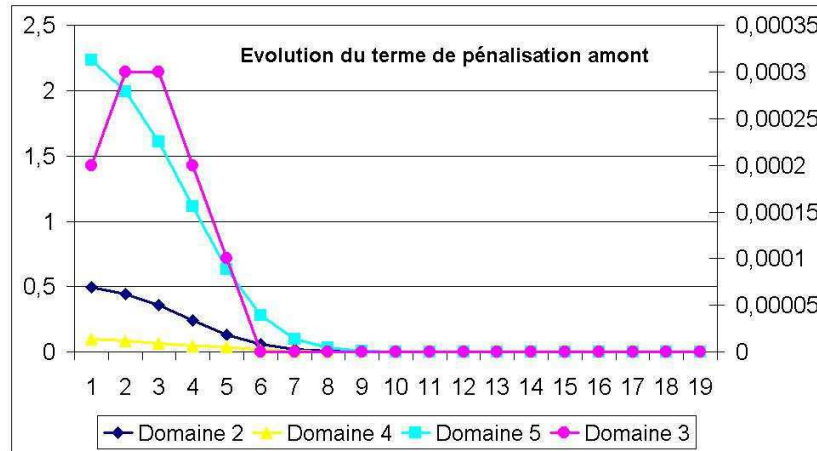


FIG. 8.13 – Illustration de l'évolution du critère d'arrêt de la méthode itérative pour l'exemple de la rivière simple. L'évolution de ce critère pour le domaine 3 est mesurée sur l'axe des ordonnées de droite alors que pour les trois autres sous domaines c'est l'axe de gauche qui mesure cette valeur.

En recoupant les figures 8.13, 8.4 et 8.5, nous voyons que l'itération 8 semble être le point où l'estimation conjointe de la condition initiale et de la condition aux limites amont est la plus fiable. Cependant, on peut essayer d'être encore plus précis en imposant un critère d'arrêt sur chaque sous domaine. Par conséquent, et en recoupant toujours les mêmes figures on peut déterminer un critère d'arrêt qui donnera une itération d'arrêt différente pour chaque sous domaine :

	Itération d'arrêt
Sous domaine 1	8
Sous domaine 2	3
Sous domaine 3	6
Sous domaine 4	8
Sous domaine 5	8

8.7 Conclusion

Le but de ce chapitre était de tester une méthode de découpage de domaine avec contrôle des interfaces entre sous domaine grâce à la méthode d'assimilation de données. Cette méthode permet d'estimer conjointement la condition initiale et les conditions aux limites sur chaque sous domaine en parallélisant le code.

Nous avons ajouté à la définition de la fonction coût une méthode itérative permettant de donner de plus en plus de poids aux termes mesurant la différence entre les valeurs aux interfaces venant de l'amont et de l'aval.

Dans un premier temps, nous avons mené une étude sur une rivière de géométrie régulière. Que ce soit avec un découpage homogène de la rivière ou en privilégiant la taille d'un sous domaine par rapport aux autres, la méthode itérative permet d'améliorer l'estimation des variables. De plus, la parallélisation du code permet un gain de temps non négligeable (facteur 2 pour 5 processeurs).

Ensuite, nous avons dégradé le jeu des observations en supposant qu'elles étaient disponibles sur tous les sous domaines excepté sur l'un d'entre eux. L'estimation devient alors de moins en moins acceptable avec l'augmentation de la taille du domaine dépourvu d'observations. Toutefois, ce résultat semble intéressant puisqu'il donne un critère de découpage de la rivière en fonction de la distribution des observations.

Enfin, un critère d'arrêt de la méthode itérative a été recherché. Ce critère basé sur la valeur des termes de pénalisation de la fonction coût permet d'arrêter les itérations au moment où l'estimation est la meilleure.

Cependant maintenant que nous avons trouvé un critère d'arrêt afin de stopper la méthode itérative, le problème de la parallélisation se pose. En effet, puisque chaque processeur s'occupe de calculer l'estimation des variables sur chaque sous domaine, le temps de calcul global va dépendre du processeur le plus lent, c'est-à-dire du sous domaine où la méthode itérative sera la plus longue. Ce problème ne peut être résolu par la diminution de la taille du sous domaine incriminé puisque rien ne prouve que la longueur de la méthode itérative dépende de cette taille.

Chapitre 9

Conclusion

Ce travail de thèse est basé sur l'étude des méthodes d'assimilation de données appliquées à l'hydraulique fluviale. La mise en oeuvre de ces méthodes nécessite un travail rigoureux afin d'écrire le code adjoint. Un outil peut permettre de faciliter ce développement de code : La différentiation automatique. Mais il ne faut pas croire que cet outil va résoudre tout les problèmes, il faut l'envisager de manière pointilleuse et respecter les nombreuses règles d'écriture de code que nous avons exposées.

De plus, les logiciels de différentiation automatique ne sont en aucun cas des compilateurs de code, c'est-à-dire qu'ils ne peuvent comprendre les fonctionnalités qui n'ont pas été envisagées par leurs concepteurs. Ainsi, à ce jour le logiciel Tapenade (ex-Odyssée), par exemple, est en cours de développement afin d'intégrer les spécificités du Fortran 95. De nombreuses discussions occasionnées par les difficultés dues aux codes dérivés dans cette thèse ont permis de déterminer avec les membres du projet TROPICS les fonctionnalités les plus nécessaires à la bonne utilisation de Tapenade.

Le logiciel de différentiation automatique Odyssée nous a aidé à dériver un logiciel professionnel d'hydraulique fluviale Carima et une maquette des écoulements bidimensionnel LMCFLD.

Nous avons testé l'estimation de la condition initiale en chaque point de calcul avec ces deux codes adjoints. Puis quelques exemples sur l'estimation de la condition aux limites avec LMCFLD ont été présentés. Et enfin, nous avons regardé le comportement d'une estimation conjointe des conditions aux limites et initiales. Dans tous les cas de figures les résultats sont bons. Mais ces exemples ont été menés avec un jeu d'observations dense en espace et en temps (une observation à chaque point de la grille et pour chaque pas de temps). Il faudrait envisager une dégradation de ce jeu de données afin de mesurer la dégradation des résultats.

De la même façon, nous avons étudié l'estimation de la rugosité en chaque point de la grille grâce au code adjoint du logiciel Carima. La difficulté de définir ce jeu de rugosité nous a incité à créer une méthode pour schématiser cette donnée. Ainsi, grâce à la minimisation d'une fonction coût utilisant un jeu d'observations dense, nous avons pu estimer un coefficient de rugosité par zones. Les quelques critères de découpage du jeu de rugosité par zones testés nous ont permis de voir qu'une grande schématisation de cette donnée ne pénalisait pas forcément la qualité des résultats.

Ensuite, nous avons considéré le problème de l'estimation de la topographie équivalente.

D'abord avec un jeu d'observation dense puis avec une dégradation progressive de cette distribution de données. Cette étude a mis en évidence la nécessité d'avoir un jeu de données en adéquation avec les caractéristiques du domaine, c'est-à-dire qu'il ne faut pas forcément miser sur la quantité des observations mais surtout sur leur localisation optimale.

Cette dernière analyse a soulevé le problème de la manière dont on peut déterminer cette distribution optimale des observations. Pour répondre à cette question, une analyse de sensibilité grâce à l'adjoint au second ordre a été effectuée. Cette analyse de la sensibilité des variables d'état à la perturbation des observations nous a permis de dresser une carte des capteurs plus ou moins indispensable.

L'analyse de sensibilité est un bon outil pour la schématisation d'un jeu de données. Ainsi, dans un premier temps nous avons essayé de schématiser un jeu de rugosité grâce à divers critères fondés sur la topographie, les essais effectués avec l'analyse de sensibilité nous ont montré que l'on pouvait rapidement découper un domaine en fonction de l'influence de la rugosité sur le calcul d'une variable d'état.

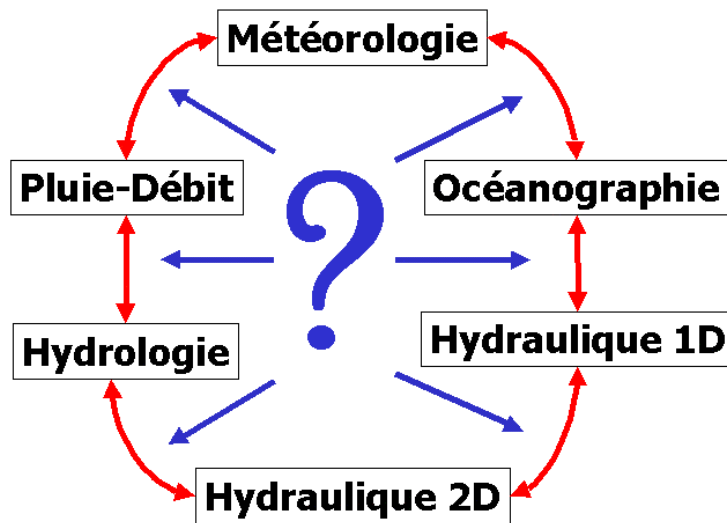


FIG. 9.1 – Illustration du problème posé par le couplage de modèle dans le cas du cycle de l'eau.

Ensuite, nous nous sommes penché sur le problème de la parallélisation à travers la décomposition de domaine. Ainsi, une rivière Ω est elle subdivisée en n sous domaines. L'écoulement sur chaque sous domaine est calculé par un code direct sur des processeurs indépendant. La parallélisation gère la communication entre les valeurs aux interfaces des sous domaines voisins. Le but de la méthode proposée est d'estimer conjointement les conditions initiales et aux limites grâce à un jeu d'observations.

A ce schéma de travail, nous avons ajouté une méthode itérative dans la définition de la fonction coût qui va permettre de donner de plus en plus de poids aux termes de pénalisation qui mesure l'écart aux interfaces entre les valeurs venant de l'amont et celle venant de l'aval. Les quelques exemples proposées tendent à montrer que cette méthode permet non seulement d'améliorer l'estimation des variables mais aussi de gagner un temps non négligeable. De plus, cette méthode semble fournir un critère pour la décomposition de domaine en

fonction de la répartition des observations disponibles.

Afin d'améliorer encore le gain de temps dû à la parallélisation, il faudrait paralléliser le code adjoint lui-même (voir la thèse de Tremolet [19]). C'est à dire que chaque processeur pourrait être remplacé par un cluster afin d'accélérer l'estimation de variables sur chaque sous domaines.

Dans un cas plus général, il faudrait tester ces méthodes de décomposition de domaine dans le cas de couplage de modèles d'origines très différentes. Ainsi, dans un modèle général cherchant à modéliser le cycle de l'eau (voir figure 9.1), il serait intéressant de voir ce que peut apporter une telle approche.

Annexe A

Quelques notions mathématiques

La dérivée directionnelle

Soit V et H deux espace de Banach.

On dit que $J : V \rightarrow H$ admet en $u \in V$ une dérivée directionnelle dans la direction v si et seulement si

$$\lim_{\theta \rightarrow 0^+} \frac{J(u + \theta v) - J(u)}{\theta} \quad \text{existe}$$

Cette dérivée est notée $J'(u, v)$.

La Gâteaux dérivée

Soit V et H deux espace de Banach.

On dit que $J : V \rightarrow H$ est Gâteaux dérivable en v si

1. $J'(u, v)$ existe $\forall v \in V$
2. $v \rightarrow J'(u, v)$ est linéaire continue par rapport à v , i.e.

$$J'(u, v) = (J'(u), v) = J'(u)(v)$$

et on note $J'(u)$ la Gâteaux dérivée.

L'intégration par parties (IPP)

si f et g sont deux fonctions continues sur l'intervalle $I = [0, 1]$ alors

$$\int_0^1 f(x).g'(x)dx = f(1).g(1) - f(0).g(0) - \int_0^1 f'(x).g(x)dx$$

Le produit scalaire

Le produit scalaire de deux vecteurs $U = (u_i)_{i=0, \dots, n}$ et $V = (v_i)_{i=0, \dots, n}$ est :

$$\langle V, U \rangle = \langle U, V \rangle = \sum_{i=0}^n u_i.v_i$$

Matrice symétrique définie positive

Soit H une matrice carrée telle que $H = (h_{ij})_{i=1,\dots,n}^{j=1,\dots,n}$ et X un vecteur de dimension n tel que $X = (x_i)_{i=1,\dots,n}$.

on dit que

- H est symétrique si

$$(h_{ij})_{i=1,\dots,n}^{j=1,\dots,n} = (h_{ji})_{i=1,\dots,n}^{j=1,\dots,n}$$

- H est définie si

$$X^t.H.X = 0 \Rightarrow X = 0$$

- H est positive si

$$X^t.H.X \geq 0$$

Bibliographie

- [1] D.G. Cacuci, P.J. Maudlin et C.V. Parks. *Adjoint sensitivity analysis of extremum-type responses in reactor safety*. Nuc. Sci. Eng., 1983.
- [2] J. A. Cunge, F. M. Holly, A. Verwey. *Practical aspects of computational river hydraulics*. Iowa institute of hydraulics research., 1980.
- [3] J. A. Cunge. *Étude d'un schéma de différences finies appliqué à l'intégration numérique d'un certain type d'équation hyperbolique d'écoulement*. Thèse préparée à la faculté de Grenoble, 1966.
- [4] J. Douglas, H. H. Rachford. *On the numerical solution of heat conduction in two and three space variables*. Trans. Amer. Math. Soc., 82 :421-439, 1956.
- [5] Faure Christelle, Papegay Yves. *Odysee User's guide Version 1.7*. INRIA Rapport technique, 1998.
- [6] Fread D.L., Lewis J.M. *NWS FLDWAV MODEL Theoretical description and user documentation*. Hydrologic reasearch laboratory report, 1998.
- [7] Hascoet Laurent. [http ://www-sop.inria.fr/tropics/tapenade.html](http://www-sop.inria.fr/tropics/tapenade.html). Web site.
- [8] F.X Le Dimet et J. Blum. *Assimilation de données pour les fluides géophysiques*. Matapli n°67, 2002.
- [9] F.X Le Dimet I.M. Navon et Dacian N. Daescu. *Second order information in data assimilation*. Month. Weat. Rev, 130, 629-648, 2002.
- [10] F.X Le Dimet, H-E Ngodock, B. Luong et J. Verron. *Sensitivity analysis in variational data assimilation*. J of the meteorological society of Japan, Vol 75 : 245-255, 1997.
- [11] F.X Le Dimet et O. Talagrand. *Variational algorithms for analysis and assimilation of meteorological observation : theoretical aspects*. Tellus, 38A : 97-110, 1986.
- [12] Lemarechal Claude. *The modules M1QN3 and N1QN3*. Version 2.0c, 1995.
- [13] Liggett et Cunge. *numerical methods of solution of the unsteady flow equations, chapter 4 of unsteady flow in open channels*. Water resources publications, fort collins, Colorado, 1975.
- [14] Martin Losch et Carl Wunsch. *Bottom topography as a control variable in an ocean model*. Journal of Atmospheric and Oceanic Technology. , 2003.
- [15] G. I. Marchuk. *On the theory of the splitting-up method..* In Numerical solutions for partial differential equations II, 1971.
- [16] Ngodock Hans Emmanuel. *Assimilation de données et analyse de sensibilité : une application à la circulation océanique*. Thèse préparée au sein du laboratoire LMC de Grenoble, 1996.
- [17] Ngnepieba Pierre. *Assimilation de données et identification de paramètres : une application en hydrologie*. Thèse préparée au sein du laboratoire LMC de Grenoble, 2001.

- [18] Preissmann A. *Propagation des intumescences dans les canaux et rivières*. First congress of the French association for computation, Grenoble, 1961.
- [19] Tremolet Yannick. *Parallélisation d'algorithmes variationnels d'assimilation de données en météorologie..* Thèse préparée au sein du laboratoire LMC de Grenoble, 1995.
- [20] Vreugdenhil. *computational methods for channel flow*. publication no 100 Delft hydraulics laboratory Delft, 1973.
- [21] E.P. Wigner. *Effects of small perturbations on pile period*. Tecnical report CP, Chicago report, 1945.
- [22] Patrick Vidard. *Vers la prise en compte des erreurs modèle en assimilation de données 4D variationnelle. Application à un modèle réaliste d'océan*. Thèse préparée au sein du laboratoire LMC de Grenoble, 2001.
- [23] N. N. Yanenko. *Méthodes des pas fractionnaires*. Armand colin, 1968, 1966.
- [24] Yang Junqing. *Assimilation de données variationnelle pour les problèmes de transport des sédiments en rivière*. Thèse préparée au sein du laboratoire LMC de Grenoble, 1999.
- [25] Q.C. Zeng, Z.Z. Ji et R.L. Li. *The construction of difference schemes for evolution equations and the numerical simulation of offshore currents*. Atmospheric sciences, pages 166-175, 1988.

Liste des tableaux

3.1	a) Exemple de représentation des coefficients de Manning dans LMCFLD b) Exemple de relief pris en compte dans LMCFLD.	61
5.1	Illustration de l'estimation de la condition initiale pour une rivière simple. a- (resp. c-) La courbe grise (point rond) désigne la valeur de la condition initiale sur le débit (resp. la hauteur d'eau) à retrouver, la courbe rouge (point carré), la valeur perturbée de la condition initiale sur le débit (resp. la hauteur d'eau) et la courbe bleue (point triangulaire), la valeur de la condition initiale sur le débit (resp. la hauteur d'eau) obtenue par la minimisation de la fonction coût. b- (resp. d-) La courbe en bleu (point carré) est l'erreur initiale sur le débit (resp. la hauteur d'eau) et la courbe en rouge (point rond) est l'erreur finale sur le débit (resp. la hauteur d'eau). Le débit est exprimé en $m^3.s^{-1}$ et la hauteur d'eau en m.	89
5.2	Illustration de l'estimation de la condition initiale pour le cas test de l'Isère. a- (resp. c-) La courbe grise (point rond) désigne la valeur de la condition initiale sur le débit (resp. la hauteur d'eau) à retrouver, la courbe rouge (point carré), la valeur perturbée de la condition initiale sur le débit (resp. la hauteur d'eau) et la courbe bleue (point triangulaire), la valeur de la condition initiale sur le débit (resp. la hauteur d'eau) obtenue par la minimisation de la fonction coût. b- (resp. d-) La courbe en bleu (point carré) est l'erreur initiale sur le débit (resp. la hauteur d'eau) et la courbe en rouge (point rond) est l'erreur finale sur le débit (resp. la hauteur d'eau). Le débit est exprimé en $m^3.s^{-1}$ et la hauteur d'eau en m.	92
5.3	Illustration de l'estimation de la condition initiale pour la cas test d'un bassin a fond plat. a- La condition initiale donnée sur la vitesse V (en $m^3.s^{-1}$). Cette condition initiale est celle que l'assimilation de donnée variationnelle doit nous aider à retrouver. b- La condition initiale en V (en $m^3.s^{-1}$) permettant de démarrer l'exécution du code d'optimisation, celle-ci est assez éloignée de celle que nous devons retrouver. c- La condition initiale en V (en $m^3.s^{-1}$) obtenue grâce à l'aide de la minimisation de la fonction coût. .	95
5.4	Illustration de l'estimation de la condition initiale pour la cas test d'un bassin a fond plat. a- La condition initiale donnée sur la hauteur d'eau Z_n (en m). Cette condition initiale est celle que l'assimilation de donnée variationnelle doit nous aider à retrouver. b- La condition initiale sur la hauteur d'eau Z_n (en m) permettant de démarrer l'exécution du code d'optimisation, celle-ci est assez éloignée de celle que nous devons retrouver. c- La condition initiale en Z_n (en m) obtenue grâce à l'aide de la minimisation de la fonction coût.	96

5.5	<i>Illustration de l'efficacité de l'estimation de la condition initiale pour la cas test d'un bassin a fond plat. a) Erreur relative commise lors de l'estimation de la condition initiale en V (en $m^3.s^{-1}$), dans le cas d'un domaine représentant un bassin à fond plat. Etude menée avec le logiciel LMCFLD.b) Erreur relative commise lors de l'estimation de la condition initiale en Z_N (en m), dans le cas d'un domaine représentant un bassin à fond plat. Etude menée avec le logiciel LMCFLD.</i>	97
5.6	<i>Illustration de l'estimation de la condition initiale pour la cas test d'un bassin dont le fond est légèrement en pente. a- La condition initiale donnée sur la hauteur d'eau, Z_n. Cette condition initiale est celle que l'assimilation de donnée variationnelle doit nous aider à retrouver. b- La condition initiale en Z_n permettant de démarrer l'exécution du code d'optimisation, celle-ci est assez éloignée de celle que nous devons retrouver. c- La condition initiale en Z_n obtenue grâce à l'aide de la minimisation de la fonction coût.</i>	98
5.7	<i>Illustration de l'estimation de la condition aux limites pour la cas test d'un bassin dont le fond est légèrement en pente. a, d et g- Les conditions aux limites à retrouver grâce à la méthode d'assimilation de données dédiée à l'estimation de la condition aux limites développée précédemment. b, e et i- Les conditions aux limites données pour initialiser l'exécution de l'adjoint. c, f et i- Les Conditions aux limites obtenues grâce à la minimisation de la fonction coût.</i>	101
5.8	<i>Illustration de l'erreur de l'estimation de la condition aux limites dans un bassin plat. Ces courbes représentent les erreurs commise par la méthode de l'adjoint afin de retrouver les conditions aux limites optimales dans les trois cas exposés précédemment.</i>	102
5.9	<i>Illustration de l'erreur de l'estimation de la condition aux limites dans un bassin plat. Ces courbes représentent les erreurs commises par la méthode de l'adjoint afin de retrouver la condition aux limites optimale suivant la pente du fond de la rivière. La condition aux limites étudiée dans ces cas est constante dans le temps.</i>	104
5.10	<i>Illustration de l'erreur de l'estimation de la condition aux limites dans un bassin plat. Ces courbes représentent les erreurs commises par la méthode de l'adjoint afin de retrouver la condition aux limites optimale suivant la pente du fond de la rivière. La condition aux limites étudiée dans ces cas est celle étudiée dans la section précédente sous la dénomination a.</i>	105
5.11	<i>Illustration de l'erreur de l'estimation conjointe de la condition aux limites et de la condition initiale dans un bassin plat. Ces courbes représentent les erreurs commises par la méthode de l'adjoint afin de retrouver les conditions aux limites (figure a) et initiale (figure b) optimales dans les trois cas exposés précédemment.</i>	107
5.12	<i>Illustration de l'estimation conjointe de la condition aux limites et de la condition initiale dans un bassin plat. a (resp b-) La courbe représente la condition aux limites (resp. initiale) à retrouver grâce à la méthode d'assimilation de données dédiée à l'estimation de la condition aux limites et de la condition initiale développée précédemment. c (resp. d-) La condition aux limites (resp. initiale) donnée pour initialiser l'exécution de l'adjoint. e (resp. f-) La Condition aux limites (resp. initiale) obtenue grâce à la minimisation de la fonction coût.</i>	108

6.1	<i>Illustration de l'estimation de la rugosité dans un bassin plat. Le coefficient de rugosité sur la digue a été choisi très grand de manière à la faire ressortir dans la représentation graphique. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est celle que l'assimilation de données variationnelle doit nous aider à retrouver. b- La rugosité permettant de démarrer l'exécution du code d'optimisation, celle-ci est assez éloignée de celle que nous devons retrouver. c- La rugosité obtenue grâce à la minimisation de la fonction coût.</i>	113
6.2	<i>Illustration de l'efficacité de l'estimation de la rugosité dans un bassin plat. a- La différence entre le jeu de rugosité initial et celui obtenu grâce à la minimisation de la fonction coût.. b- Evolution de la valeur de la fonction coût (6.1) en fonction des itérations du code d'optimisation.</i>	114
6.3	<i>Illustration de l'estimation de la rugosité pour un jeu de Manning binaire. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est celle que l'assimilation de données variationnelle doit nous aider à retrouver. b- La rugosité permettant de démarrer l'exécution du code d'optimisation, celle-ci est assez éloignée de celle que nous devons retrouver. c- La rugosité obtenue grâce à la minimisation de la fonction coût.</i>	116
6.4	<i>Illustration de l'efficacité de l'estimation de la rugosité pour un jeu de Manning binaire. a- La différence entre le jeu de rugosité initial et celui obtenu grâce à la minimisation de la fonction coût.. b- Evolution de la valeur de la fonction coût (6.1) en fonction des itérations du code d'optimisation.</i>	117
6.5	<i>Illustration de l'estimation de la rugosité pour le cas test de la colline. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est celle que l'assimilation de données variationnelle doit nous aider à retrouver. b- La rugosité permettant de démarrer l'exécution du code d'optimisation, celle-ci est assez éloignée de celle que nous devons retrouver. c- La rugosité obtenue grâce à la minimisation de la fonction coût.</i>	119
6.6	<i>Illustration de l'efficacité de l'estimation de la rugosité pour le cas test de la colline. a- La différence entre le jeu de rugosité initial et celui obtenu grâce à la minimisation de la fonction coût.. b- Evolution de la valeur de la fonction coût (6.1) en fonction des itérations du code d'optimisation.</i>	120
6.7	<i>Illustration de l'estimation de la rugosité pour un jeu de Manning très hétérogène. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est celle que l'assimilation de données variationnelle doit nous aider à retrouver. b- La rugosité permettant de démarrer l'exécution du code d'optimisation, celle-ci est assez éloignée de celle que nous devons retrouver. c- La rugosité obtenue grâce à la minimisation de la fonction coût.</i>	122
6.8	<i>Illustration de l'efficacité de l'estimation de la rugosité pour un jeu de Manning très hétérogène. a- La différence entre le jeu de rugosité initial et celui obtenu grâce à la minimisation de la fonction coût.. b- Evolution de la valeur de la fonction coût (6.1) en fonction des itérations du code d'optimisation.</i>	123
6.9	<i>Illustration de l'estimation de la rugosité par zone. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est celle que l'assimilation de données variationnelle doit nous aider à approcher. b- La rugosité permettant de démarrer l'exécution du code d'optimisation, celle-ci est assez éloignée de celle que nous devons retrouver. c- La rugosité obtenue grâce à la minimisation de la fonction coût.</i>	125

6.10	<i>Illustration de l'estimation de la rugosité par zone avec une fonction coût mesurant l'erreur commise sur le calcul de la hauteur d'eau. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est celle que l'assimilation de données variationnelle doit nous aider à approcher. b- La rugosité permettant de démarrer l'exécution du code d'optimisation, celle-ci est assez éloignée de celle que nous devons retrouver. c- La rugosité obtenue grâce à la minimisation de la fonction coût.</i>	127
6.11	<i>Présentation du domaine Loire. a- Représentation du Val d'Ouzouer en amont d'Orléans sur la Loire. La zone entourée de rouge est le domaine considéré dans les exemples présentés ci dessous. b- Le relief d'une partie du Val d'Ouzouer sur la Loire. c- La représentation de la rugosité sur une partie du val d'Ouzouer (SPOT).</i>	129
6.12	<i>Illustration de l'estimation de la rugosité par zone pour le domaine Loire grâce à un critère de découpage fonction de l'altitude. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est celle que l'assimilation de données variationnelle doit nous aider à approcher. b- La rugosité permettant de démarrer l'exécution du code d'optimisation, celle-ci est assez éloignée de celle que nous devons retrouver. c- La rugosité obtenue grâce à la minimisation de la fonction coût.</i>	131
6.13	<i>Illustration de l'estimation de la rugosité par zone pour le domaine Loire grâce à un critère de découpage fonction de la pente. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est celle que l'assimilation de données variationnelle doit nous aider à approcher. b- La rugosité permettant de démarrer l'exécution du code d'optimisation, celle-ci est assez éloignée de celle que nous devons retrouver. c- La rugosité obtenue grâce à la minimisation de la fonction coût.</i>	132
6.14	<i>Illustration de l'estimation de la topographie équivalente pour le cas test de la colline. a- La topographie du bassin. Ce relief est celui que l'assimilation de données variationnelle doit nous aider à retrouver. b- Le relief permettant de démarrer l'exécution du code d'optimisation, celui-ci est assez éloigné de celui que nous devons retrouver. c- La topographie obtenue grâce à l'aide de la minimisation de la fonction coût.</i>	135
6.15	<i>Illustration de l'efficacité de l'estimation de la topographie équivalente pour le cas test de la colline. a- La différence entre la topographie initiale et celle obtenu grâce à la minimisation de la fonction coût. b- Evolution de la valeur de la fonction coût (6.7) en fonction des itérations du code d'optimisation.</i>	136
6.16	<i>Illustration de l'estimation de la topographie équivalente pour le cas test de la colline avec une fonction coût ne mesurant l'erreur que sur les hauteurs d'eau. a- La topographie du bassin. Ce relief est celui que l'assimilation de données variationnelle doit nous aider à retrouver. b- Le relief permettant de démarrer l'exécution du code d'optimisation, celui-ci est assez éloigné de celui que nous devons retrouver. c- La topographie obtenue grâce à l'aide de la minimisation de la fonction coût.</i>	137
6.17	<i>Illustration de l'efficacité de l'estimation de la topographie équivalente pour le cas test de la colline. a- La différence entre la topographie initiale et celle obtenu grâce à la minimisation de la fonction coût. b- Evolution de la valeur de la fonction coût (6.7) en fonction des itérations du code d'optimisation.</i>	138

6.18	<i>Illustration de l'estimation de la topographie équivalente pour le cas test de la route avec un jeu d'observations complet. a- La topographie du bassin. Ce relief est celui que l'assimilation de données variationnelle doit nous aider à retrouver. b- Le relief permettant de démarrer l'exécution du code d'optimisation, celui-ci est assez éloigné de celui que nous devons retrouver. c- La topographie obtenue grâce à l'aide de la minimisation de la fonction coût.</i>	140
6.19	<i>Illustration de l'efficacité de l'estimation de la topographie équivalente pour le cas test de la route avec un jeu d'observations complet. a- La différence entre la topographie initiale et celle obtenu grâce à la minimisation de la fonction coût. b- Evolution de la valeur de la fonction coût (6.9) en fonction des itérations du code d'optimisation.</i>	141
6.20	<i>Illustration de l'estimation de la topographie équivalente pour le cas test de la route avec un jeu d'observations partout sauf au dessus de la route. a- La topographie du bassin. Ce relief est celui que l'assimilation de données variationnelle doit nous aider à retrouver. b- Le relief permettant de démarrer l'exécution du code d'optimisation, celui-ci est assez éloigné de celui que nous devons retrouver. c- La topographie obtenue grâce à l'aide de la minimisation de la fonction coût.</i>	143
6.21	<i>Illustration de l'efficacité de l'estimation de la topographie équivalente pour le cas test de la route avec un jeu d'observations partout sauf au dessus de la route. a- La différence entre la topographie initiale et celle obtenu grâce à la minimisation de la fonction coût. b- Evolution de la valeur de la fonction coût (6.9) en fonction des itérations du code d'optimisation.</i>	144
6.22	<i>Illustration de l'estimation de la topographie équivalente pour le cas test de la route avec un jeu d'observations partout sauf autour de la route. a- La topographie du bassin. Ce relief est celui que l'assimilation de données variationnelle doit nous aider à retrouver. b- Le relief permettant de démarrer l'exécution du code d'optimisation, celui-ci est assez éloigné de celui que nous devons retrouver. c- La topographie obtenue grâce à l'aide de la minimisation de la fonction coût.</i>	145
6.23	<i>Illustration de l'efficacité de l'estimation de la topographie équivalente pour le cas test de la route avec un jeu d'observations partout sauf autour de la route. a- La différence entre la topographie initiale et celle obtenu grâce à la minimisation de la fonction coût. b- Evolution de la valeur de la fonction coût (6.9) en fonction des itérations du code d'optimisation.</i>	146
6.24	<i>Illustration de l'estimation de la topographie équivalente pour le cas test de la route avec un jeu d'observations uniquement disponible au dessus de la route. a- La topographie du bassin. Ce relief est celui que l'assimilation de données variationnelle doit nous aider à retrouver. b- Le relief permettant de démarrer l'exécution du code d'optimisation, celui-ci est assez éloigné de celui que nous devons retrouver. c- La topographie obtenue grâce à l'aide de la minimisation de la fonction coût.</i>	147
6.25	<i>Illustration de l'efficacité de l'estimation de la topographie équivalente pour le cas test de la route avec un jeu d'observations uniquement disponible au dessus de la route. a- La différence entre la topographie initiale et celle obtenu grâce à la minimisation de la fonction coût. b- Evolution de la valeur de la fonction coût (6.9) en fonction des itérations du code d'optimisation. . . .</i>	148

6.26	<i>Illustration de l'estimation de la topographie équivalente pour le cas test de la Loire. a- La topographie du bassin. Ce relief est celui que l'assimilation de données variationnelle doit nous aider à retrouver. b- Le relief permettant de démarrer l'exécution du code d'optimisation, celui-ci est assez éloigné de celui que nous devons retrouver. c- La topographie obtenue grâce à l'aide de la minimisation de la fonction coût.</i>	150
6.27	<i>Illustration de l'efficacité de l'estimation de la topographie équivalente pour le cas test de la Loire. a- La différence entre la topographie initiale et celle obtenu grâce à la minimisation de la fonction coût. b- Evolution de la valeur de la fonction coût (6.10) en fonction des itérations du code d'optimisation.</i>	151
7.1	<i>a- Sensibilité de la prévision sur la vitesse U par rapport à la perturbation du coefficient de Manning au point (20,18). b- Carte des sensibilités de la prévision de la vitesse U à la perturbation des coefficients de Manning sur le domaine Loire.</i>	154
7.2	<i>Illustration de l'utilisation de la sensibilité pour la schématisation du jeu de Manning en 7 zones de Manning équivalent. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est la rugosité réelle. b- La rugosité permettant de démarrer l'exécution du code d'optimisation. c- La rugosité obtenue grâce de la minimisation de la fonction coût.</i>	156
7.3	<i>Illustration de l'utilisation de la sensibilité pour la schématisation du jeu de Manning en 19 zones de Manning équivalent. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est la rugosité réelle. b- La rugosité permettant de démarrer l'exécution du code d'optimisation. c- La rugosité obtenue grâce de la minimisation de la fonction coût.</i>	157
7.4	<i>Illustration de l'utilisation de la sensibilité pour la schématisation du jeu de Manning en 7 zones de Manning équivalent. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est la rugosité réelle. b- La rugosité permettant de démarrer l'exécution du code d'optimisation. c- La rugosité obtenue grâce de la minimisation de la fonction coût.</i>	159
7.5	<i>Illustration de l'utilisation de la sensibilité pour la schématisation du jeu de Manning en 23 zones de Manning équivalent. a- La rugosité du bassin (coefficient de Manning). Cette rugosité est la rugosité réelle. b- La rugosité permettant de démarrer l'exécution du code d'optimisation. c- La rugosité obtenue grâce de la minimisation de la fonction coût.</i>	162
8.1	<i>Relief du lit de la rivière (a) et distribution de la rugosité (b) dans le cas test illustrant la méthode de décomposition de domaine avec un exemple plus complexe.</i>	173

Table des figures

1.1	<i>Illustration de l'effet de l'assimilation de données variationnelle sur le calcul de la trajectoire du modèle. Dans la figure de gauche la courbe discontinue (bleue) décrit la trajectoire idéale du modèle et la courbe continue (rouge) décrit la trajectoire calculée par le modèle, ces deux dernières sont assez éloignées. La figure de droite montre qu'après minimisation, i.e. mise en place de la méthode de l'assimilation de données, la trajectoire calculée rejoint la trajectoire idéale. Les cercles verts désignent la position des observations disponibles. La "trajectoire" est l'évolution de l'état du système (les valeurs des variables d'état en tout point du domaine) en fonction du temps.</i>	12
1.2	<i>Illustration de l'importance de l'estimation de la condition initiale. Dans la figure de gauche, la courbe continue (bleue) est la trajectoire idéale et la courbe discontinue (rouge) est la trajectoire calculée. Après estimation d'une condition initiale optimale, la figure de droite illustre le rapprochement entre la trajectoire calculée et idéale.</i>	13
1.3	<i>Illustration du problème posé par la détermination d'un jeu de Manning optimal à partir d'une distribution des rugosités complexe. Comment ne pas perdre l'information contenue dans les mailles?</i>	13
1.4	<i>Illustration du problème posé par la détermination d'une topographie optimale à partir d'une topographie du domaine plus précise que celle nécessaire. Comment ne pas perdre l'information contenue dans les mailles?</i>	14
1.5	<i>Illustration du problème posé pour obtenir une topographie bidimensionnelle de la rivière à partir de la connaissance des quelques sections.</i>	15
1.6	<i>Illustration de l'utilité de la méthode de l'assimilation de données variationnelle pour le couplage de modèles d'hydraulique fluvial 1d et 2D.</i>	15
3.1	<i>Visualisation de la modélisation de la Loire dans le Val d'Ouzouer par le logiciel Carima.</i>	36
3.2	<i>Représentation schématique du découpage d'une plaine inondable en casier.</i>	37
3.3	<i>Visualisation de la modélisation de la plaine inondable couplée à celle de la Loire dans le Val d'Ouzouer par le logiciel Carima.</i>	37
3.4	<i>Schéma implicite de Preissmann utilisé dans le logiciel Carima.</i>	39
3.5	<i>Organigramme du code de calcul Carima.</i>	45
3.6	<i>Topographie mise en relief par les lignes de niveaux décrivant le val d'Ouzouer.</i>	46
3.7	<i>Description d'une coupe transversale avec le logiciel Carima.</i>	46
3.8	<i>Découpage de La Loire en coupe transversales et sections hydrauliques avec le code de calcul Carima.</i>	47
3.9	<i>Découpage du Val d'Ouzouer en casiers avec le code de calcul Carima.</i>	48
3.10	<i>Modélisation du Val d'Ouzouer avec le code de calcul Carima.</i>	48
3.11	<i>Scénarii utilisés dans l'étude du Val d'Ouzouer avec Carima.</i>	49

3.12	<i>Inondation du Val d'Ouzouer pour un scénario de période de retour 200 ans modélisé avec le code de calcul Carima.</i>	49
3.13	<i>Composition du domaine Ω considéré par LMCFLD.</i>	52
3.14	<i>Distribution des variables dans un maillage de type ARAKAWA. Cette distribution est utilisée pour discrétiser les équations de Saint Venant dans le logiciel LMCFLD.</i>	54
3.15	<i>Construction d'un domaine non rectangulaire modélisé par LMCFLD.</i>	61
3.16	<i>Organigramme du code de calcul LMCFLD</i>	62
4.1	<i>Organigramme du code de calcul CARIMA-SOGREAH. La partie de ce code qui doit être dérivée est mise en relief.</i>	65
4.2	<i>Arborescence classique d'un code de calcul. L'écriture du code adjoint requiert de savoir comment parcourir le code direct dans le sens de la remontée.</i>	71
5.1	<i>Cas test de la rivière simple pour le logiciel Carima.</i>	90
5.2	<i>Cas test de la rivière simple traité par le code de calcul Carima : évolution de la fonction coût en fonction de l'itération du code de minimisation.</i>	90
5.3	<i>Représentation de l'Isère entre Pontcharra et Grenoble. Ce cas réel est utilisé dans le cadre de l'estimation de la condition initiale avec le logiciel Carima.</i>	91
5.4	<i>Cas test de l'Isère pour le code de calcul Carima : évolution de la fonction coût en fonction de l'itération du code de minimisation.</i>	93
5.5	<i>Représentation du bassin dont le fond est en pente légère.</i>	97
5.6	<i>Erreur commise lors de l'estimation de la condition initiale donnée en ZN sur le domaine représentant un bassin dont le fond est en pente. Étude menée avec le logiciel LMCFLD.</i>	99
6.1	<i>Evolution de la valeur de la fonction coût (6.4) en fonction des itérations du code d'optimisation.</i>	124
6.2	<i>Dans la colonne Fonction coût : C'est l'évolution de la valeur de la fonction coût (6.5) en fonction des itérations du code d'optimisation.</i>	126
6.3	<i>Evolution de la valeur de la fonction coût (6.6) en fonction des itérations du code d'optimisation.</i>	130
6.4	<i>Dans la colonne Fonction coût : C'est l'évolution de la valeur de la fonction coût (6.6) en fonction des itérations du code d'optimisation.</i>	133
7.1	<i>Carte des sensibilités de la prévision de la vitesse V (en $m^3.s^{-1}$) à la perturbation des coefficients de Manning sur le domaine Loire.</i>	155
7.2	<i>Carte des sensibilités de la prévision des hauteurs d'eau par rapport à la perturbation des coefficients de Manning sur le domaine Loire.</i>	158
7.3	<i>Carte de la sensibilité de la prévision des vitesses horizontales par rapport aux observations sur un domaine simple à fond plat. Vue de dessus.</i>	160
7.4	<i>Carte de la sensibilité de la prévision des hauteurs d'eau par rapport aux observations sur un domaine à fond plat au centre duquel se trouve une route.</i>	161
8.1	<i>Problème posé par le couplage d'un modèle monodimensionnel avec un modèle bidimensionnel.</i>	163
8.2	<i>Subdivision de la rivière Ω en n sous domaines.</i>	165
8.3	<i>Condition aux limites imposée en amont dans le cas test illustrant la méthode de décomposition de domaine.</i>	167

8.4	<i>Normes de l'erreurs relatives commises sur les 5 sous domaines et à chaque itération k afin de retrouver la condition initiale pour le cas test de la rivière avec un jeu d'observations complet. La norme utilisée ici est la norme de la matrice des erreurs relatives commises sur le domaine (les éléments de la matrice répertorient les erreurs relatives commises en chaque point de la grille).</i>	168
8.5	<i>Normes de l'erreurs relatives commises sur les 5 sous domaines et à chaque itération k afin de retrouver la condition aux limites amont pour le cas test de la rivière avec un jeu d'observations complet.</i>	169
8.6	<i>Speed Up obtenu sur la plateforme Mirage pour le cas test de la rivière avec un jeu d'observations complet.</i>	169
8.7	<i>Normes de l'erreurs relatives commises sur les 5 sous domaines et à chaque itération k afin de retrouver la condition initiale pour le cas test de la rivière avec un jeu d'observations dégradé.</i>	170
8.8	<i>Normes de l'erreurs relatives commises sur les 5 sous domaines et à chaque itération k afin de retrouver la condition aux limites amont pour le cas test de la rivière avec un jeu d'observations dégradé.</i>	170
8.9	<i>Normes de l'erreurs relatives commises sur les 5 sous domaines et à chaque itération k afin de retrouver la condition initiale pour le cas test de la rivière avec un jeu d'observations dégradé et un sous domaine numéro 3 de taille disproportionné.</i>	171
8.10	<i>Normes de l'erreurs relatives commises sur les 5 sous domaines et à chaque itération k afin de retrouver la condition aux limites amont pour le cas test de la rivière avec un jeu d'observations dégradé un sous domaine numéro 3 de taille disproportionné.</i>	172
8.11	<i>Normes de l'erreurs relatives commises sur les 5 sous domaines et à chaque itération k afin de retrouver la condition initiale pour le cas test de la rivière "chaotique" avec un jeu d'observations dégradé et un sous domaine numéro 3 de taille disproportionné.</i>	173
8.12	<i>Normes de l'erreurs relatives commises sur les 5 sous domaines et à chaque itération k afin de retrouver la condition aux limites amont pour le cas test de la rivière "chaotique" avec un jeu d'observations dégradé et un sous domaine numéro 3 de taille disproportionné.</i>	174
8.13	<i>Illustration de l'évolution du critère d'arrêt de la méthode itérative pour l'exemple de la rivière simple. L'évolution de ce critère pour le domaine 3 est mesurée sur l'axe des ordonnées de droite alors que pour les trois autres sous domaines c'est l'axe de gauche qui mesure cette valeur.</i>	175
9.1	<i>Illustration du problème posé par le couplage de modèle dans le cas du cycle de l'eau.</i>	178

Assimilation de données pour les modèles d'hydraulique fluviale.
Estimation de paramètres, analyse de sensibilité et décomposition de
domaine.

Mots-clefs

Assimilation de données variationnelle, système adjoint, estimation de paramètres, analyse de sensibilité, décomposition de domaine, parallélisation.

Résumé

Le calcul de prévisions fiables des inondations ne peut se concevoir sans l'utilisation de toutes les informations disponibles. Les équations mathématiques qui permettent la construction d'un modèle, les images satellites par exemple qui décrivent le domaine, mais aussi les mesures faites *in situ*. Ces observations sont nécessaires à la bonne qualité des prévisions puisqu'elles sont le seul témoin de la réalité du terrain.

Le but de ce mémoire est de présenter et de tester des méthodes fondées sur le contrôle optimal permettant une utilisation optimale de toutes les informations disponibles. Diverses applications de ces méthodes seront présentées : l'estimation de paramètres, l'analyse de sensibilité et le couplage de modèles.

Data assimilation for hydraulic models.
Parameters estimation, sensitivity analysis and domain decomposition.

Key-words

variational data assimilation, adjoint system, parameters estimation, sensitivity analysis, domain decomposition, parallelization.

Abstract

Flood prediction requires to use every available information. Mathematical equations which allows to build a model, satellite images describing the domain and *in situ* measurement. Those observations are necessary to make a good prediction because they are the witness of the terrain reality.

The aim of this work is to present and test some methods based on the control theory which allows to use optimally every available informations. Some applications of these methods will be treated as parameters estimation, sensitivity analysis and coupling models.