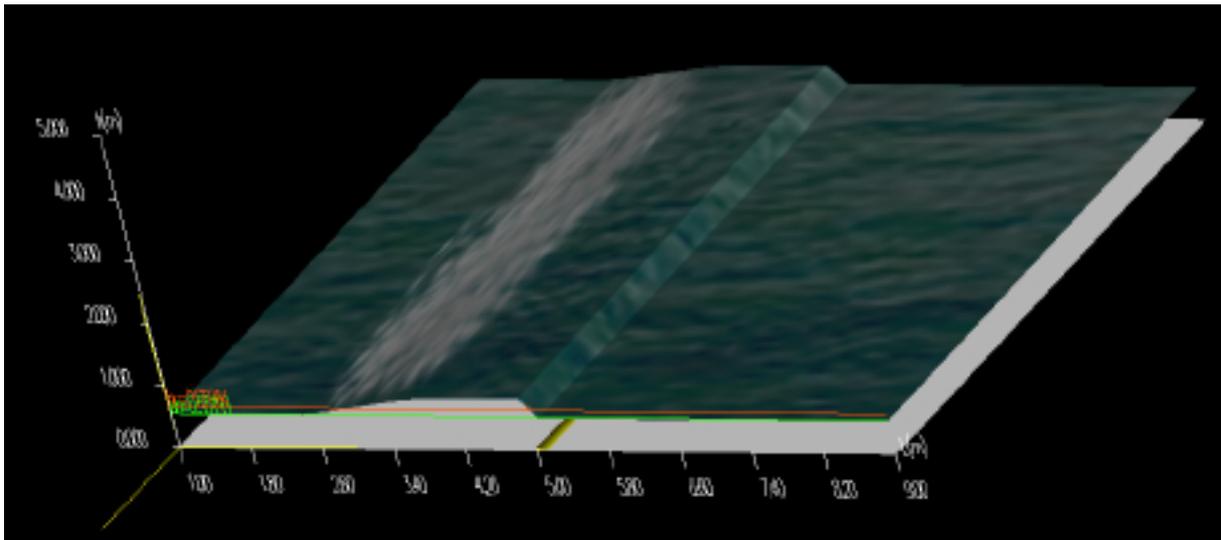


Année Spéciale Informatique ENSIMAG

## Simulation d'un ruisseau par approches phénoménologiques pour la synthèse d'images



Cédric DODARD

Stage effectué au sein de l'équipe iMAGIS du laboratoire GRAVIR sous la responsabilité de Fabrice NEYRET

Juin 2001



## **Remerciements**

Je remercie Fabrice Neyret mon maître de stage pour m'avoir accueilli au sein de l'équipe iMAGIS, ainsi que pour l'attention qu'il a porté à mes travaux.

Je remercie également toutes les personnes avec lesquelles j'ai pu avoir des contacts au sein du laboratoire, et qui m'ont apporté leur aide.



# Sommaire

## 1 Introduction

1.1 Le projet iMAGIS .....	1
1.1.1 Présentation .....	1
1.1.2 Thèmes de recherche .....	1
1.2 But du stage : simulation d'un ruisseau .....	2
1.2.1 Contexte .....	2
1.2.1 Sujet du stage .....	3

## 2 Hypothèses, présentation du problème

2.1 Hypothèses.....	4
2.2 Présentation des éléments clés.....	5
2.2.1 Le nombre de Froude .....	5
2.2.3 Charge spécifique, profondeur normale, profondeur critique .....	5

## 3 Le calcul

3.1 Les différents régimes.....	6
3.2 Le calcul des paramètres.....	8
3.2.1 La profondeur normale $h_n$ .....	8
3.2.2 La profondeur critique $h_c$ .....	8
3.2.3 La charge spécifique $H_s$ .....	8
3.2.4 Expression de $h(x)$ pour le graduellement varié .....	9
3.2.5 Le ressaut.....	9
3.3 La méthode de calcul .....	10
3.3.1 Cas avec transition / ressaut .....	10
3.3.2 Cas du régime stable .....	12

## 4 Le programme

4.1 L'affichage : OpenGL .....	13
4.2 La structure.....	13
4.3 L'animation .....	14

## 5 Les résultats : les 4 régimes.....

17

## Conclusion et perspectives .....

19

## Bibliographie .....

21

## Annexes : Code du programme .....

22



# 1 Introduction

## 1.1 Le projet iMAGIS

L'équipe iMAGIS est une équipe du laboratoire GRAVIR-IMAG (liant le CNRS, l'INPG et l'UJF). Le projet iMAGIS est un projet de l'INIRA. C'est au sein de cette équipe que j'ai effectué mon stage.

### 1.1.1 Présentation

Le projet iMAGIS concerne le domaine de l'informatique graphique et de la synthèse d'images.

iMAGIS développe des outils permettant de concevoir, puis d'utiliser dans le cadre d'applications de tailles significatives, en particulier en vue de simulation, des maquettes numériques 3D. Ces maquettes peuvent être purement géométriques ou posséder des propriétés physiques.

Les applications visées se situent dans des domaines très divers (construction automobile ou aéronautique, urbanisme, éclairagisme, bâtiment, téléphonie mobile, chimie, chirurgie assistée, agronomie, environnement, audiovisuel, video-ludique etc...).

Dans bien des cas, il s'agit de concevoir les techniques (modélisation et algorithmiques graphiques) sur lesquelles reposent les systèmes de "réalité virtuelle" qui commencent à voir le jour. Le défi à relever est de leur fournir la puissance nécessaire à l'affichage et à l'interaction "temps réel" qui les caractérisent.

### 1.1.2 Thèmes de recherche

Visualisation d'environnements complexes

Rendu réaliste, simulation de l'éclairage (utilisation de modèles complexes de réflectance, éclairage global, techniques de radiosité hiérarchiques, environnements dynamiques)

Animation, modélisation d'objets déformables, modélisation de leur comportement (modèles à base de surfaces implicites, simulation du mouvement, détection et réponse à des collisions, fortes déformations, manipulation interactive)

Algorithmique de la visibilité, structures de données efficaces pour le rendu de scènes très complexes (prise en compte de la cohérence, cas dynamique)

Interactivité, réalité augmentée (manipulation et partage d'objets virtuels, intégration image-son)

## 1.2 But du stage : simulation d'un ruisseau

### 1.2.1 Contexte

La simulation des phénomènes naturels intéresse depuis longtemps les chercheurs en synthèse d'image. Leur richesse visuelle découle souvent d'une grande complexité de comportement et/ou d'apparence, c'est ce qui engendre leur intérêt applicatif, la motivation de ceux qui les étudient, et le défi scientifique que constitue leur simulation.

Parmi ces phénomènes naturels figurent en bonne place les fluides, et en particulier l'eau.

La synthèse d'image a deux buts principaux

La création d'images interactives, utilisées par exemple pour les jeux et les simulateurs

La création d'images réalistes pour les effets spéciaux, l'architecture, le design...

Dans le premier cas, les images doivent être "belles", mais ce qui importe avant tout est de les créer en "temps réel", c'est à dire tous les 25<sup>ème</sup> de seconde.

Dans le second cas, les images doivent être "les plus belles possibles", et on accepte couramment une demie heure de calcul par image, voire plusieurs heures pour une image unique.

Bien que de façon générale, l'objectif des techniques de synthèse d'images soit de produire des comportements et des aspects plausibles, et non quantitativement précis, une tendance courante est de recourir aux équations de la physique, et aux méthodes de résolution numérique pour les simuler, profitant ainsi des avancées d'autres disciplines en matière de simulation.

Dans le cas des fluides, cette démarche peut vite devenir coûteuse dans le cas de résolutions 3D des équations de la mécanique des fluides, d'autant plus que les scènes naturelles requièrent souvent à la fois des domaines larges (paysages...) et une résolution fine (importance des détails visibles pour un rendu réaliste).

Par ailleurs, bien souvent, seule l'interface du fluide avec le milieu environnant (la surface d'un ruisseau par exemple) nous intéresse. Dès lors, pourquoi simuler "l'intérieur" du milieu ?

En outre, une modélisation exhaustive au sens physique du fluide ne permet pas à l'utilisateur graphiste de contrôler facilement le rendu de la scène (à savoir de donner l'aspect qu'il souhaite, amplitude, position des différents phénomènes), ce qui est pourtant une des doléances essentielles de la synthèse d'image.

Par conséquent, afin d'éviter toutes ces difficultés, on adoptera une approche phénoménologique basée sur la modélisation de la surface libre (surface de l'eau du ruisseau) au niveau des perturbations locales introduites (à savoir ici, un seuil : par ex. une pierre immergée barrant le lit sur toute sa largeur) .

Ce stage fait suite à celui de Nathalie Praizelin (avril-septembre 2000) qui avait modélisé les ondes et perturbations générées au niveau de la surface de l'eau par des obstacles (pierres) introduits dans le cours d'une rivière.

### **1.2.2 Sujet du stage**

Comme précisé précédemment, ce stage fait suite à celui de Nathalie Praizelin, le tout s'inscrivant dans un projet plus vaste, la simulation complète de l'apparence d'un ruisseau sous ses aspects visuels et temporels. On s'intéresse en fait simultanément à la forme de la surface, à son animation, et au rendu visuel qui en découle.

Si le sujet "Simulation d'un ruisseau par approches phénoménologiques pour la synthèse d'images" est assez vaste, étant donnée la durée relativement courte de ce stage (1 mois), on va en fait s'intéresser à la forme de la surface libre dans un cas particulier. Pour les différents régimes : torrentiel, fluvial, on s'intéressera à une tranche longitudinale d'un canal de section rectangulaire dans lequel un obstacle (dénommé "seuil" ) est immergé sur toute la largeur du canal. Les calculs seront menés dans le cas stationnaire, à savoir qu'une fois les paramètres de l'écoulement fixés, la forme de la surface libre sera déterminée et restera fixe, son animation (l'eau qui "coule") n'étant confiée qu'à l'affichage (pas au calcul), et n'étant de fait qu'un apport visuel destiné à embellir le modèle.

Cette étude bien que représentant un cas particulier (notamment du fait de l'obstruction du ruisseau sur toute la largeur : seuil) devrait permettre par la suite de modéliser en réelle 3D (à l'aide d'une grille adaptée) la surface d'un ruisseau avec des pierres immergées localement.

## 2 Hypothèses, présentation du problème

### 2.1 Hypothèses

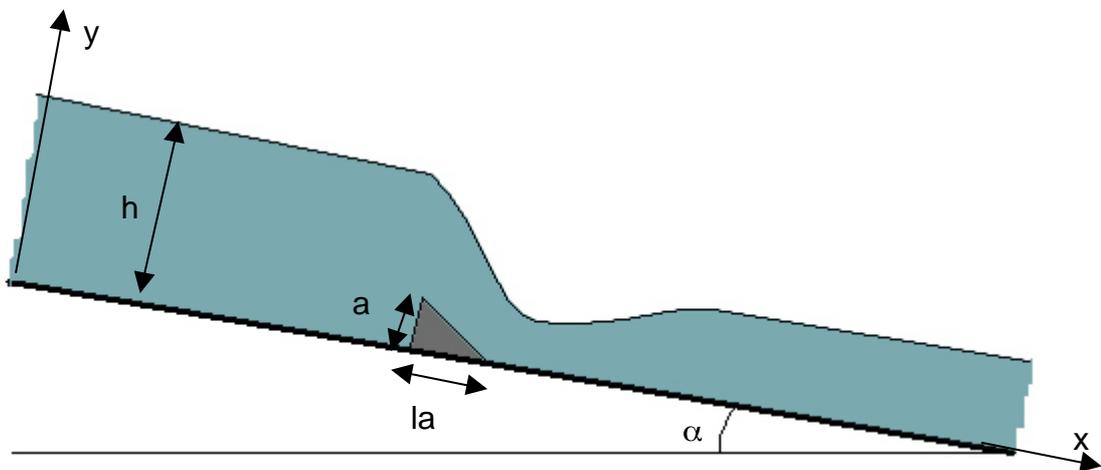
Tous nos calculs seront effectués en régime permanent ( $\frac{f}{ft} = 0$ ), dans un canal de section rectangulaire. Le problème se présente comme un problème 2D, c'est pourquoi nous traiterons une coupe longitudinale, qui pourra être extrudée pour un rendu 3D.

Dans nos écoulements, on supposera les profils de vitesses uniformes (écoulement turbulent) et les vecteurs vitesses parallèles entre eux (et donc parallèles au fond) dans les parties uniformes et graduellement variées.

Par ailleurs, le seuil est de petite dimension (longueur, hauteur) par rapport aux dimensions caractéristiques de l'écoulement, si bien qu'il sera considéré comme un obstacle local et pas une rupture de pente. En outre on ne considérera aucune perte de charge au niveau du seuil proprement dit.

*Précision concernant la bibliographie : l'ensemble des formules explicitées dans ce rapport, ainsi que les divers schémas se retrouvent dans les deux manuels d'hydraulique générale [1] et [2].*

La figure ci-dessous présente la coupe utilisée dans les calculs.



On définit :

- h, profondeur par rapport au fond (m)
- a, hauteur du seuil (m)
- la, longueur du seuil (m)
- s0, pente par rapport à l'horizontale =  $\tan(\alpha)$
- Q, débit volumique ( $m^3 / s$ )
- b, largeur (m)
- q, débit unitaire =  $\frac{Q}{b}$
- V, vitesse (m/s) =  $\frac{Q}{b \cdot h}$

## 2.2 Présentation des éléments clés

### 2.2.1 Le nombre de Froude

Soit  $Fr$  le nombre de Froude :  $Fr = \frac{V}{\sqrt{g \cdot h}}$

Si  $Fr < 1$ , on a un écoulement fluvial (les ondes peuvent remonter le courant, l'amont est sensible à ce qui se passe en aval)

Si  $Fr > 1$ , on a un écoulement torrentiel (l'amont est indépendant de ce qui se passe à l'aval)

Cette notion de Froude est importante, car en fluvial, l'amont étant sensible à l'aval, les calculs de la ligne d'eau devront être effectués avec les conditions limites aval. En régime fluvial, la surface libre sera donc calculée de l'aval vers l'amont, c'est à dire à contre courant.

En torrentiel, c'est l'amont qui "commande" l'écoulement, la surface libre sera donc obtenue de l'amont vers l'aval.

On voit donc bien l'importance de cette notion pour l'algorithme de calcul qui nous permettra d'obtenir la surface de notre ruisseau.

### 2.2.3 Charge spécifique, profondeur normale, profondeur critique

On appelle charge spécifique  $H_s = h \cdot \cos(\alpha) + \frac{V^2}{2 \cdot g}$  la charge par rapport au fond du canal.

On appelle profondeur normale  $h_n$  la profondeur en régime uniforme établi (c'est à dire la profondeur d'une rivière ayant travaillé son lit, lorsqu'on ne la perturbe pas).

On appelle profondeur critique  $h_c$  la profondeur qui correspond au débit maximum pour une charge spécifique donnée. Cette profondeur correspond à Froude=1 soit :  $h = h_c \quad Fr = 1$

L'intérêt vient du fait qu'à partir de ces grandeurs, on peut caractériser le régime d'écoulement et ensuite déterminer la surface libre.

### 3 Le calcul

#### 3.1 Les différents régimes

Comme on vient de le dire,  $h_n, h_c, H_s$  permettent de savoir dans quel cas d'écoulement on se situe et donc de mener les calculs correspondants.

On a  $Fr = \frac{V}{\sqrt{g \cdot h}} = \frac{q}{h \sqrt{g \cdot h}}$ , donc le régime fluvial à  $Fr < 1$  correspond à un écoulement uniforme (loin du seuil) à  $h = h_n > h_c$  ( $h = h_c$   $Fr = 1$ ). De la même façon, un écoulement torrentiel correspond à un écoulement uniforme à  $h = h_n < h_c$ .

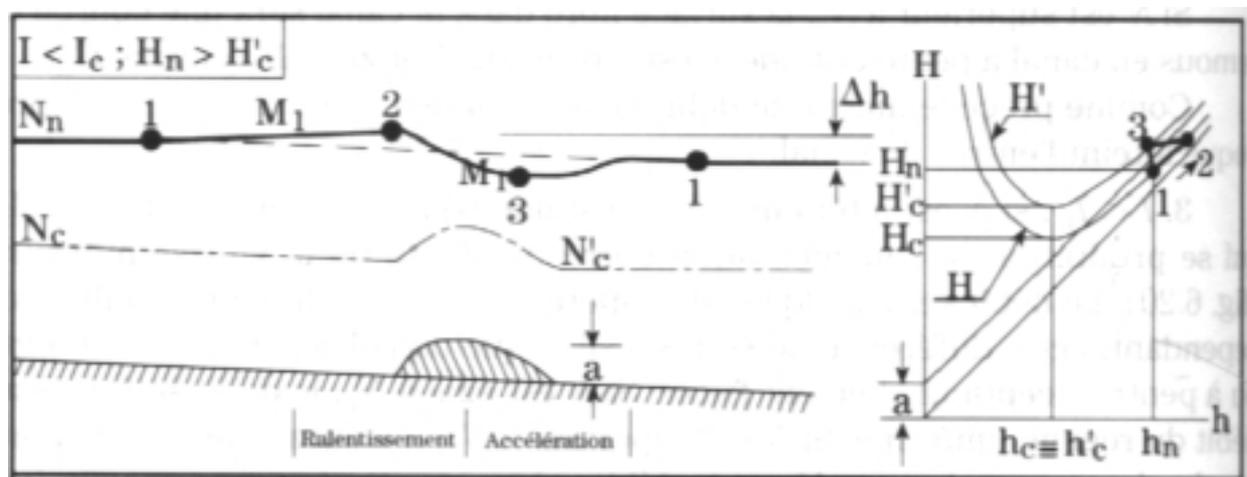
Par ailleurs, une fois que l'on a identifié le régime "loin" à l'amont du seuil, il reste encore à savoir si une transition entre les deux régimes se produira au niveau de l'obstacle. La charge spécifique  $H_s$  va alors nous permettre de savoir si il y aura apparition d'un ressaut près du seuil (et donc apparition de remous importants, voire d'écume) ou si l'on restera stable, avec conservation du régime d'origine (on aura alors qu'une légère perturbation de la surface libre).

On définit  $H_n = H_s(h_n)$  la charge spécifique normale, et  $H_c = H_s(h_c) + a \cdot \cos \alpha = H_s(h_c) + a \cdot \cos \alpha$  ( $h_c = h_c$ ) la charge spécifique critique au sommet du seuil.

Si  $H_n > H_c$  on n'a pas de transition, on reste dans le régime amont, mais si  $H_n < H_c$  alors on aura une transition entre les régimes et la formation d'un ressaut près du seuil.

On obtient donc quatre cas à traiter. Ceux-ci sont explicités dans les graphes ci-après tirés de [1] *l'hydraulique générale de A. Lancaster*.

$h_n > h_c$  et  $H_n > H_c$  (fluvial en amont, pas de transition)





## 3.2 Le calcul des paramètres

Nous allons maintenant nous intéresser au calcul des différents paramètres nécessaires pour trouver la forme de la surface libre.

### 3.2.1 La profondeur normale $h_n$

Comme on a déjà pu le préciser, la profondeur normale correspond à la profondeur en régime uniforme. C'est par exemple la profondeur d'une rivière qui a "travaillé" son lit et atteint un équilibre.

La formule de Chezy caractérise les écoulements uniformes et permet donc de tirer  $h_n$  connaissant les autres paramètres.

Chezy donne :  $Q = C A \sqrt{R_H \sin \alpha}$

avec Q : débit volumique  
C : coefficient de Chezy ( $30 < C < 100$ ),  
plus le canal est lisse, plus C est élevé !

$A = b h$  : Section "mouillée"

$R_H = \frac{b h}{2 b + h}$  : rayon hydraulique (canal rectangulaire)

$\alpha$  : angle du canal avec l'horizontale

Soit finalement (écoulement uniforme  $h = h_n$ ) :

$$h_n^3 - \frac{2 Q^2}{C^2 b^3 \sin \alpha} h_n - \frac{Q^2}{b^2 C^2 \sin \alpha} = 0 \quad ? \quad h_n$$

### 3.2.2 La profondeur critique $h_c$

La profondeur critique correspond à Froude  $Fr=1$ , c'est donc du nombre de Froude qu'on tire sa valeur :

$$Fr = 1 = \frac{V}{\sqrt{g h_c}} = \frac{Q}{b h_c \sqrt{g h_c}} \quad ? \quad h_c = \sqrt[3]{\frac{Q^2}{g b^2}}$$

### 3.2.3 La charge spécifique $H_s$

La charge spécifique correspond à la charge par rapport au fond du canal et s'exprime donc en fonction de h de la manière suivante :

$$H_s = h \cos(\alpha) + \frac{V^2}{2g} = h \cos \alpha + \frac{Q^2}{2 b^2 h^2 g}$$

Au sommet du seuil (de hauteur  $a$ ), la charge spécifique est exprimée à partir de la charge spécifique du canal par :

$$H_s = H_s + a \cos \alpha = (h + a) \cos \alpha + \frac{Q^2}{2 b^2 \gamma h^2 \gamma g}$$

### 3.2.4 Expression de $h(x)$ pour le graduellement varié

Lorsqu'on perturbe l'écoulement, il tend à se ramener à une position d'équilibre, à une profondeur d'équilibre qui finalement correspondra à  $h = h_n$ . La subtilité vient du fait que selon le régime, déterminé par la profondeur  $h$  connue dans une section dite de contrôle, on doit mener le calcul soit de l'amont vers l'aval, soit de l'aval vers l'amont. La section est dite de contrôle car elle sert de base au calcul sachant qu'elle possède des conditions limites connues.

En effet, comme on l'a déjà dit, en fluvial (dès que  $h > h_c$ ), c'est l'aval qui influence l'amont, par conséquent on doit "lire" la ligne d'eau à contre courant, c'est à dire de l'aval vers l'amont. En fluvial on calculera donc cette ligne d'eau de l'aval vers l'amont.

En torrentiel, c'est le contraire, l'amont est indépendant de l'aval, et c'est même l'amont qui influence l'aval. Par conséquent, on partira d'une section de contrôle amont vers une section aval pour le calcul.

Quoi qu'il en soit, la profondeur  $h(x)$  ( $x$  porté sur le fond du canal) s'écrit de la même façon quelque soit le régime, torrentiel ou fluvial :

$$\frac{dh}{dx} = \sin \alpha \frac{h^3 - \frac{Q^2 \gamma (b + 2 \gamma h)}{C^2 \gamma b^3 \gamma \sin \alpha}}{h^3 - \frac{Q^2}{b^2 \gamma g}} \quad (\text{vient de la conservation de la quantité de mouvement})$$

Soit, une fois discrétisé :  $h = h_i + (x - x_i) \sin \alpha \frac{h_i^3 - \frac{Q^2 \gamma (b + 2 \gamma h_i)}{C^2 \gamma b^3 \gamma \sin \alpha}}{h_i^3 - \frac{Q^2}{b^2 \gamma g}}$

Ainsi, en écoulement graduellement varié, on calculera les  $h(x)$  à partir de la formule précédente soit grâce à des  $(x_i, h_i)$  amont (torrentiel), soit grâce à des  $(x_i, h_i)$  aval (fluvial).

### 3.2.5 Le ressaut

Le ressaut correspond à la transition du torrentiel vers le fluvial. C'est un écoulement dit rapidement varié dans lequel on ne peut pas appliquer les méthodes et formules du graduellement varié du fait du non respect des hypothèses d'un régime "graduel" (à savoir : profil de vitesse uniforme et parallélisme des vecteurs vitesses).

Néanmoins, connaissant une des deux profondeurs caractéristiques : soit la profondeur avant le ressaut  $h_1$  (en torrentiel) soit la profondeur après le ressaut  $h_2$  (fluvial), on est capable de déterminer l'autre,  $h_1$  et  $h_2$  étant des profondeurs dites conjuguées.

La relation de "conjugaison" est la suivante :

$$\frac{h_2}{h_1} = \frac{1}{2 \cos \alpha} \sqrt{\frac{8 Fr_1^2 \cos^3 \alpha}{1 - 2 K \tan \alpha} + 1} - 1$$

avec K donné par :

$s_0 = \tan \alpha$	0.04	0.08	0.12	0.16	0.2	0.24	0.28	0.3
K	3.2	2.7	2.3	2.0	1.8	1.6	1.4	1.35

En outre, la longueur du ressaut est calculée suivant la formule empirique suivante :

$$L_r = 6 (h_2 - h_1)$$

Ainsi, pour modéliser le ressaut, on voit bien que l'on n'a que des informations sur les hauteurs d'eau amont et aval. Par conséquent on "joindra" les deux par une spline (polynôme de degré 3) qui en première approximation, avec une texture de "remous importants" (plusieurs textures en fait, constituant une animation) donnera un rendu de ressaut acceptable.

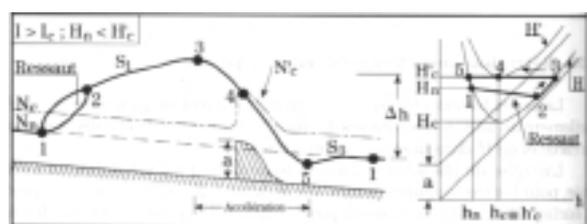
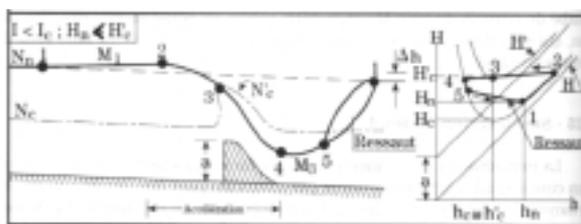
*Remarque : en ce qui concerne l'autre partie rapidement variée : la transition fluvial vers torrentiel, on ne calculera rien sur cette partie (on ne peut rien calculer !), on joindra les sections amont et aval de la même manière, par une spline (polynôme de degré 3) afin d'avoir une ligne "fluide" qualitativement acceptable.*

### 3.3 La méthode de calcul

On va maintenant s'intéresser à la façon de calculer la ligne d'eau pour les différents cas. Ce qui se dégage d'ores et déjà c'est que le problème se décompose en deux parties : d'une part les régimes ne comportant pas de transition, d'autre part les régimes avec transition et ressaut.

C'est le calcul de  $H_n$ ,  $H_c$ ,  $h_n$  et  $h_c$  (comme on l'a vu en 3.1) et leur comparaison qui va permettre de déterminer dans quel cas on se trouve et donc d'orienter le calcul.

#### 3.3.1 Cas avec transition / ressaut



Le point central de ce calcul est la connaissance de la profondeur dans la section correspondant au sommet du seuil. Dans cette section on est à la profondeur critique  $h_c$ . C'est cette section de contrôle qui dirige le calcul.

Par ailleurs, comme on déjà pu le préciser, le seuil n'introduit pas de perte de charge. Par conséquent, on considère qu'on a les mêmes énergies spécifiques dans les sections de contrôles en amont et en aval du seuil, sections 2,3,4 pour le cas fluvial (figure de gauche), sections 3,4,5 pour le cas torrentiel (figure de droite). On connaît donc les profondeurs dans ces sections car celles-ci sont toutes à l'énergie spécifique critique du seuil :  $H_c$ .

Les inconnues ici, sont, du fait du régime rapidement varié (pas de calcul de  $\frac{dh}{dx}$  possible), les abscisses respectives des sections en amont et en aval du seuil. On émettra donc l'hypothèse que celles-ci sont situées de part et d'autre du seuil à "la\*1.1" de "pa" (position du sommet du seuil), soit 110% de la longueur du seuil de chaque côté.

Comme décrit avant, la ligne d'eau entre ces sections correspond à une spline.

Dans les parties de l'écoulement où celui-ci est graduellement varié, la ligne d'eau est calculée par itération grâce à la formule donnée en (3.2.4).

Cela dit, les conditions limites diffèrent pour ces quatre courbes ( $M_1, M_3, S_1, S_3$ ) :

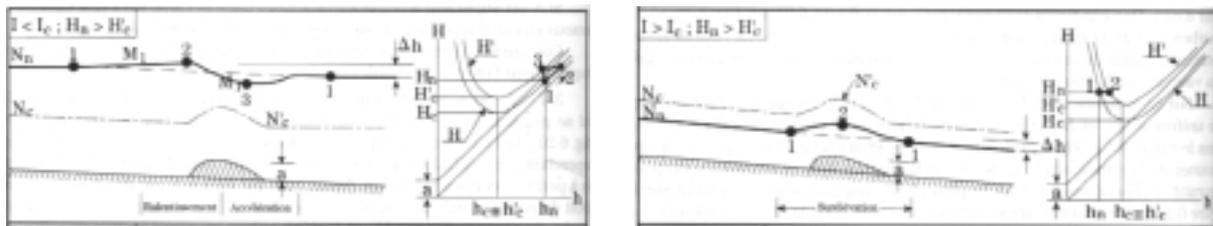
Pour  $M_1$  : on est en fluvial, on part donc de la section aval où l'on connaît  $x$  et  $h$ , et on itère jusqu'à retrouver la profondeur normale  $h_n$  à l'amont, profondeur d'équilibre du régime fluvial.

Pour  $M_3$  : on sait que le ressaut qui suit cette courbe revient à l'équilibre du régime fluvial, c'est à dire que pour ce ressaut on aura  $h_2 = h_n$ . On détermine alors  $h_1$ , profondeur amont du ressaut, et on calcule  $M_3$  par itérations depuis la section juste après le seuil, jusqu'à retrouver  $h_1$  plus loin à l'aval (on est en effet en torrentiel,  $h < h_c$ , donc on calcule dans le sens du courant). On trace ensuite une spline pour le ressaut, connaissant les profondeurs conjuguées et par conséquent la longueur du ressaut.

Pour  $S_1$  : on sait que le ressaut qui précède la courbe démarre de la profondeur d'équilibre du régime torrentiel  $h_1 = h_n$ . De la même façon que pour  $M_3$ , on calculera  $h_2$ , ce qui permettra de tracer  $S_1$  par itération depuis une section juste avant le seuil jusqu'à tomber sur  $h_2$  plus en amont (on remonte en effet le courant car  $S_1$  est en fluvial). On tracera ensuite le ressaut à cette position.

Pour  $S_3$  : on est en torrentiel, on calcule donc la ligne d'eau de l'amont vers l'aval jusqu'à  $h_n$ , profondeur du régime uniforme.

### 3.3.2 Cas du régime stable



Ici, la grosse difficulté est que l'on ne possède pas de section de contrôle principale (ou tout est connu) comme dans les cas précédents. On va donc prendre des hypothèses fortes pour pouvoir tracer une ligne d'eau descente.

Ainsi on considère un seuil, toujours sans perte de charge, avec une charge spécifique au niveau de ce dernier égale à la charge spécifique normale de l'écoulement. Grâce à cette considération on détermine la profondeur au sommet du seuil.

Autre hypothèse, on considère que la profondeur normale  $h_n$  est "atteinte" de part et d'autre du seuil à "2\*la" de son centre (position pa). On relie ensuite le tout par des splines pour avoir, en fluvial, un "nid de poule" et en torrentiel, une "bosse".

Le problème de ces deux régimes est qu'on ne connaît pour ainsi dire aucun paramètre physique pour le calcul, on est donc obligé "d'improviser" quelque peu pour avoir un rendu plausible de la surface libre.

*Remarque : de façon globale (régimes avec transition ou sans), les hypothèses les plus "fortes" sont formulées sur la position des sections de contrôles juste avant et après le seuil (abscisses par rapport au fond du canal). Probablement laissera-t-on plus tard un certain contrôle sur ces positions à l'utilisateur graphiste dans le but d'adoucir ou non la courbe de remous sur le seuil.*

## 4 Le programme

La programmation s'est faite en C, l'extension des fichiers en C++ n'étant présente que pour permettre d'inclure dans le code, du code objet, comme par exemple le code nécessaire à la création des boites de dialogue de GLUI.

### 4.1 L'affichage : OpenGL

L'affichage graphique se fait à l'aide d'OpenGL qui nous fournit toute une panoplie de primitives permettant de modéliser des objets, de les placer / déplacer dans l'espace, d'y plaquer des textures etc..., bref de dessiner notre scène en temps réel.

La gestion du fenêtrage et les interactions clavier / souris sont réalisées par GLUT. La partie "boite de dialogue" où l'on a accès au divers paramètres physiques et autres paramètres d'affichage est, elle, gérée par GLUI. GLUT et GLUI sont deux extensions d'OpenGL.

Une remarque importante concerne l'objet "ruisseau avec seuil (+axes)" qui est en fait le seul composant de notre scène : afin de ne pas déformer cet objet lors de l'inclinaison du canal après un zoom soit en X, soit en Y par exemple, l'astuce est de garder des axes orthonormés, et donc de "tricher" sur l'affichage. En effet, on ne déforme pas les axes (et donc le modèle), ceux-ci sont toujours de longueur fixée. Par contre, on affiche les points et les valeurs, multipliés par un coefficient tenant compte de l'échelle actuelle selon X et Y dans l'espace fixe des axes (on ne trace donc pas les points à leur position réelle dans l'espace 3D).

De cette façon notre ensemble est toujours "physiquement" orthonormé, mais visuellement on a effectivement un zoom, le tout avec un espace d'affichage restant toujours le même (espace défini par des axes "finis et fixes").

### 4.2 La structure

Le programme est défini autour d'un fichier principal "seuil.cpp" où se fait le calcul et l'affichage, bref le rendu de la scène. Les fichiers annexes contiennent les fonctions de calcul de la surface libre, les fonctions de résolution des équations du 3<sup>ème</sup> degré et autres chargeurs de texture.

Le calcul est indépendant de l'affichage, une fois les paramètres de l'écoulement fixés, la surface libre est stockée dans une liste chaînée. Celle-ci sera parcourue à chaque rendu (soit parce qu'on déplace l'objet pour voir la scène sous un autre angle, soit parce que l'animation réclame un rendu régulier).

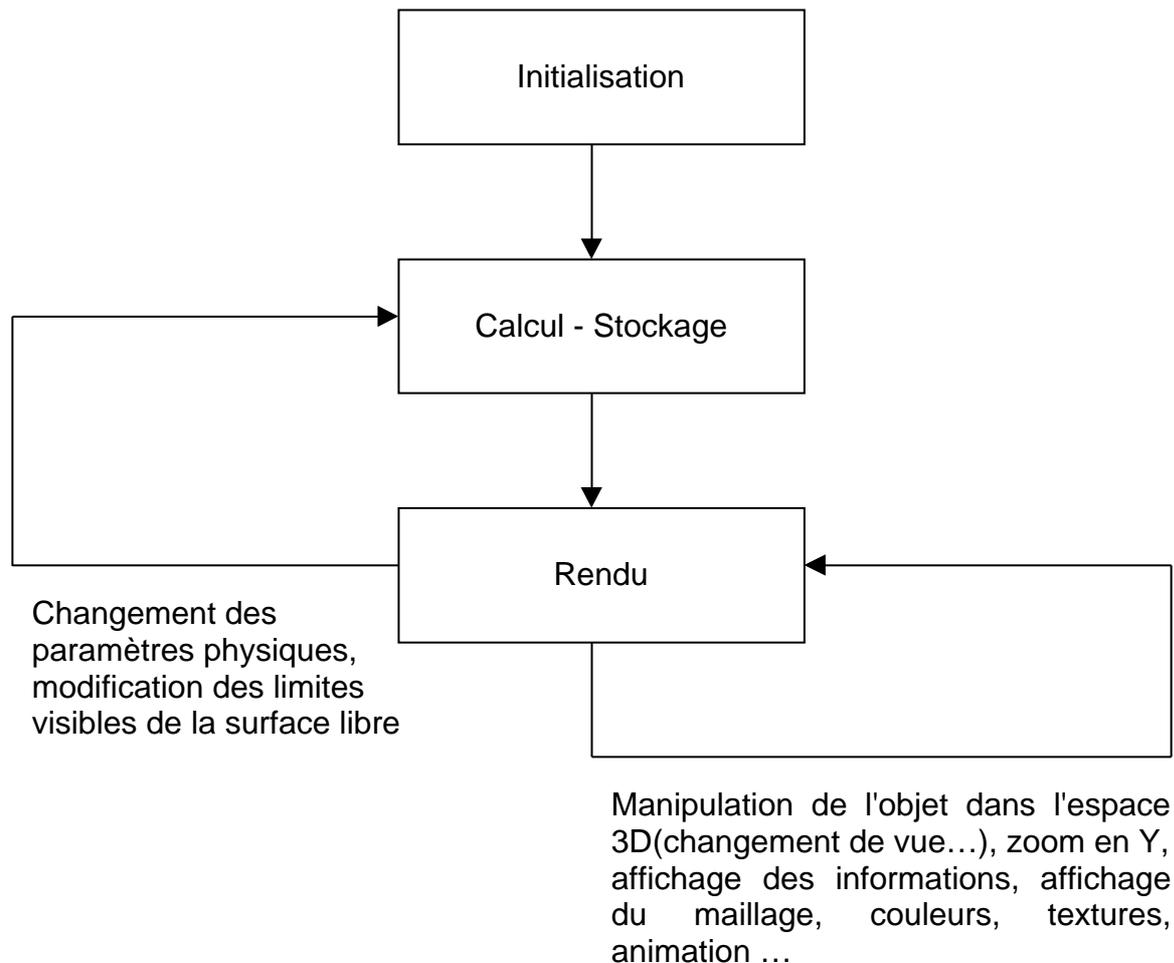
En outre, on ne calcule que les points visibles (c'est à dire ceux contenus dans l'espace défini par nos axes), sachant que le seuil est toujours visible. On ne peut, en effet, pas le "sortir" de l'espace visible car il est la base de la détermination de la surface libre.

Le re-calcul de la surface (et donc la modification de la liste chaînée) n'intervient que lorsque l'on modifie soit :

les paramètres physiques de l'écoulement.

soit les limites de la surface libre visible (zoom selon X, translation selon X ou changement dans la longueur du canal observée), il faut alors recalculer les nouveaux points visibles.

Le schéma est donc le suivant :

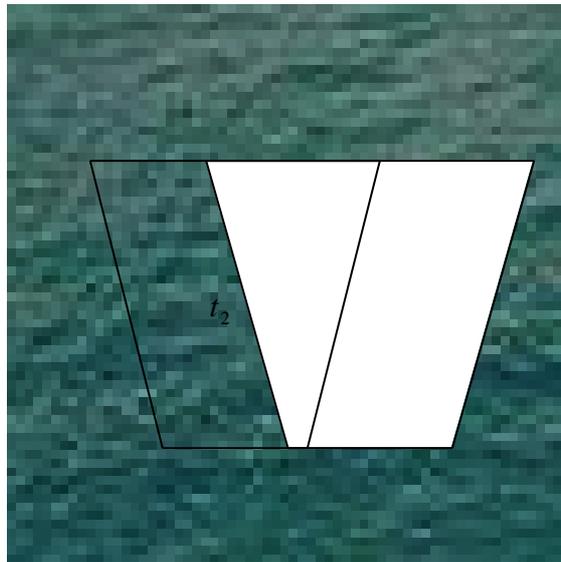


### 4.3 L'animation

Comme on déjà pu le préciser, le problème que nous avons traité ici est stationnaire. Ainsi, une fois l'écoulement calculé, la surface libre ne change plus tant qu'on ne "bouge" pas un des paramètres physiques. Cependant, dans la réalité, si la forme de la surface de l'eau est à peu près constante dans le temps, on a bien un mouvement de transport, et on observe bien le courant du ruisseau. Par conséquent, pour rendre le modèle visuellement attractif (ce qui est tout de même la finalité), il faut plaquer une texture d'eau qui bouge pour donner l'impression de transport.

En fait, le plaquage de la texture se fait par application des polygones sur une texture rectangulaire. Ainsi, pour habiller un objet, on définit ses coordonnées dans l'espace 2D de la texture, comme si on le couchait sur une "tapisserie" afin d'appliquer celle-ci. Par conséquent, un mouvement apparent de texture (et donc d'eau) s'obtient en

faisant des rendus successifs tels que l'on applique à chaque fois sur la texture l'objet translaté d'un certain déplacement constant dans le sens contraire du déplacement apparent.



Comme on le voit sur la figure ci-dessus, pour donner un mouvement de gauche à droite à la texture appliquée à notre trapèze, l'idée est de translater au temps  $t_2 > t_1$  le trapèze dans l'espace de la texture, et ainsi de suite, toujours d'un vecteur constant vers la gauche (coordonnée en abscisse du genre  $x = -V \cdot t$  dans l'espace de la texture). Ainsi, au niveau du rendu, on aura une texture mobile de gauche à droite sur notre objet, et ce, à la vitesse  $V$ .

Or, cette technique marche bien quand on veut simuler un déplacement à vitesse constante d'une texture mappée de la même façon sur tout un objet. Cependant, dans le cas de notre ruisseau avec un seuil, le débit est constant, mais la profondeur, elle, change en fonction de l'abscisse. De fait, physiquement on n'a pas les mêmes vitesses partout. Si on applique la même technique que ci-dessus, avec des vitesses locales différentes (donc des vecteurs de translation différents), on obtient une texture plus ou moins mouvante (sorte de stries) mais avec des discontinuités, ce qui visuellement (et physiquement) ne représente pas la réalité.

En fait, l'astuce consiste à considérer l'espace  $(u, v)$  de la texture non plus comme un espace en distance, mais comme un espace en temps pour le "collage" de la texture. En effet, il est nécessaire de faire ressortir les accélérations et décélérations de la surface libre pour avoir un rendu correct. Or ceci ne peut pas être fait de façon "dynamique" comme précédemment,  $V$  n'étant pas constant sur la longueur de la ligne d'eau (on doit conserver l'intégrité de la texture). C'est pour cela que l'espace  $(u, v)$  doit être vu ici comme un espace en temps. En outre, lorsqu'on réfléchit au problème, une fois le régime établi, la forme du volume d'eau du ruisseau est déterminée et ne change plus. Par conséquent, si l'on trouve la forme de la texture au niveau du 1<sup>er</sup> rendu, l'animation ne consistera qu'à translater tous les points de l'objet d'un vecteur unique.

"Physiquement", la texture devra être étirée là où la vitesse augmente et contractée

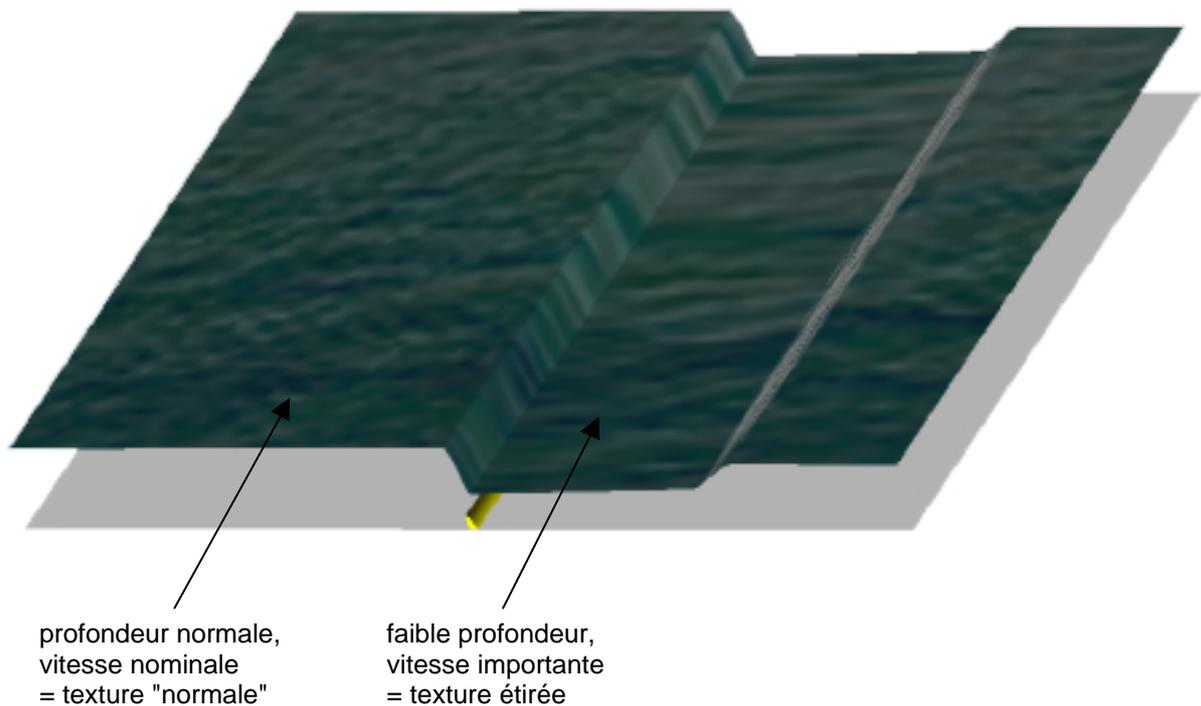
là où la vitesse diminue pour simuler la déformation de la veine fluide due aux variations de vitesse.

Dès lors, l'abscisse des points de la surface libre dans l'espace  $(u, v)$  de la texture sera du genre :  $\frac{x_{\text{sur la surface libre}}}{V}$  avec  $x_{\text{sur la surface libre}}$  la position du point sur la surface libre et  $V$  la vitesse de l'écoulement au point considéré, si bien qu'à la fin, la texture sera parfaitement appliquée à toute la surface libre avec un étirement là où  $V$  est fort et une contraction là où  $V$  est faible.

En fait, cela revient à dire que la position d'un point de la surface sur la texture correspond au temps que mettrait une particule fluide pour arriver à l'endroit correspondant sur la ligne d'eau. Ceci fixe donc la forme du plaquage de texture.

*Remarque : on divise chaque  $u$  ainsi obtenu par le  $u_{\text{max}}$  correspondant à la somme de tout les  $u$  ( $u_{\text{max}}$  aura été calculé lors d'une itération préalable). Ainsi on mappe la texture sur toute la surface (l'espace  $(u, v)$  allant de  $(0,0)$  à  $(1,1)$ ), évitant ainsi une répétition.*

On obtient alors le résultat suivant :



Physiquement ce mappage de texture est assez intuitif si l'on imagine une tache de peinture, par exemple, transportée par le ruisseau, on aura bien étirement quand la vitesse augmente et agrégation quand la vitesse diminue.

Ceci concernait l'apparence de la texture mappée. En ce qui concerne l'animation macroscopique de cette texture, à savoir le mouvement de transport de

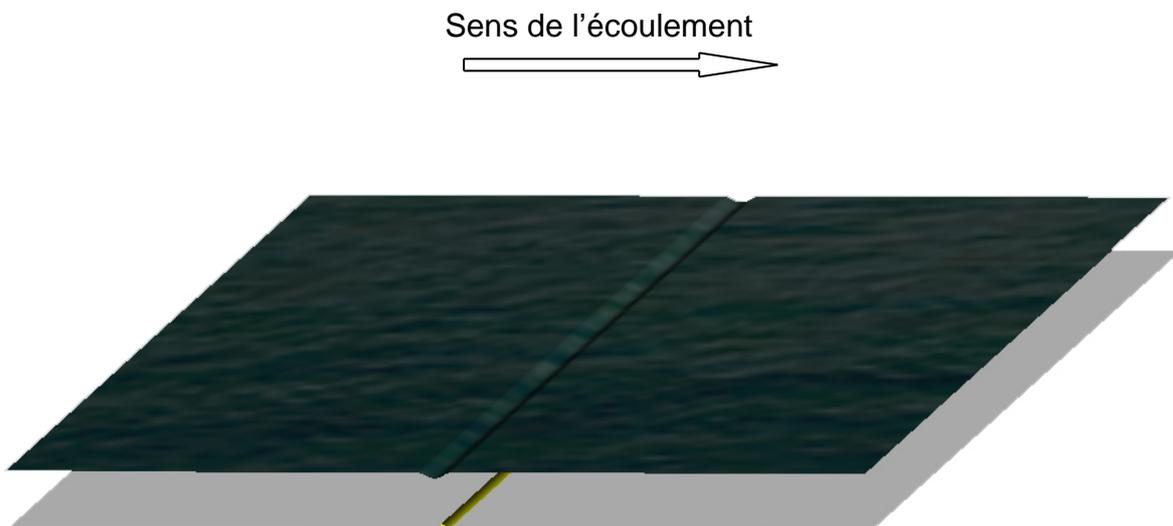
l'eau, on va se baser sur la vitesse nominale, à savoir la vitesse de l'écoulement uniforme en amont. Par conséquent au terme de temps précédent ( $\frac{x_{sur\ la\ surface\ libre}}{V}$ ) fixant la forme "statique" de la texture, on retranchera un vecteur permettant de simuler le mouvement de l'eau. Ce vecteur sera fonction de la vitesse nominale  $V_n$  de l'écoulement uniforme loin à l'amont et s'écrira :  $-V_n \cdot \text{temps}$  avec "temps" le temps itératif de l'animation (fonction idle de GLUT).

Ainsi avec des coordonnées en abscisse du type ( $\frac{x_{sur\ la\ surface\ libre}}{V}$ ) -  $V_n \cdot \text{temps}$  on aura, à la fois les accélérations / décélérations en fonction de la profondeur ainsi que le débit (mouvement macroscopique) du ruisseau en fonction des paramètres physiques. Même si les "u" de l'espace texture ne sont pas homogènes (du "temps"+ des "distances"), il nous permettent d'obtenir un rendu visuel satisfaisant.

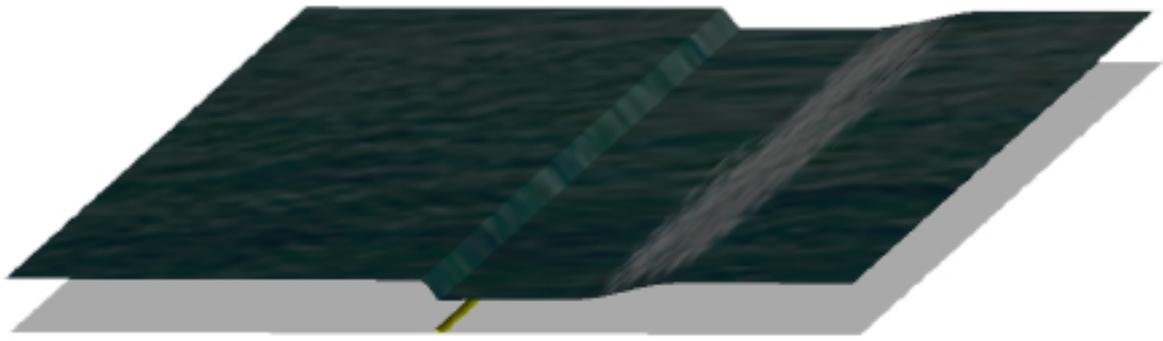
## 5 Les résultats : les 4 régimes

La transition entre les différents régimes se fait essentiellement par manipulation du débit du ruisseau et de sa pente par rapport à l'horizontale.

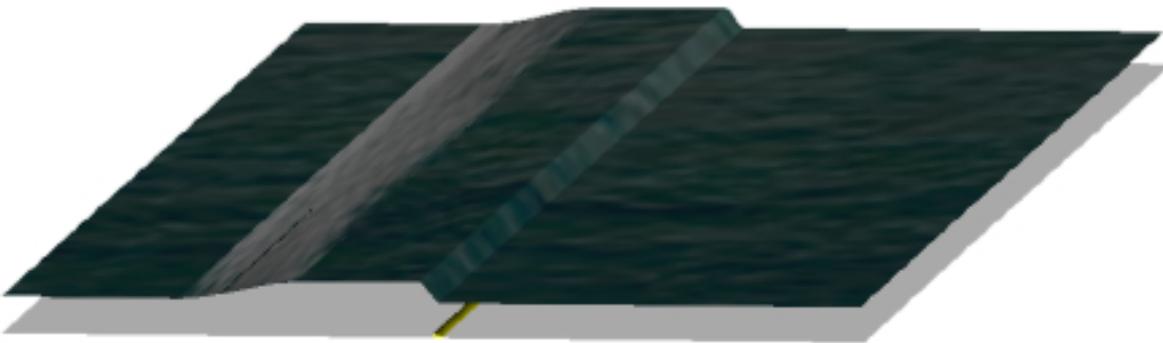
Régime fluvial, pas de transition (stable)



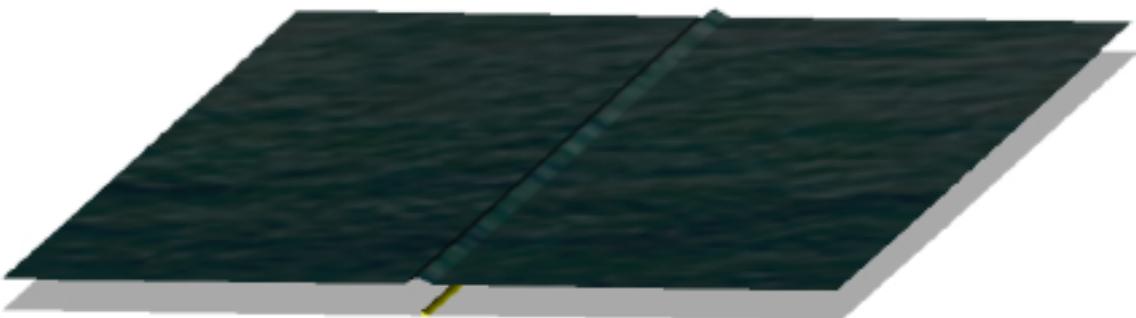
Régime fluvial avec de transition et ressaut



Régime torrentiel avec ressaut et transition



Régime torrentiel, pas de transition (stable)



## Conclusion et perspectives

On dispose à ce stade d'une modélisation 2D (extrudée en 3D) d'un ruisseau dont le passage est gêné par un seuil sur toute sa largeur. Visuellement, bon nombre d'améliorations sont possibles, néanmoins au niveau de la modélisation purement physique, le traitement de cet objet immergé semble tout à fait raisonnable, et le rendu graphique reste acceptable, notamment avec l'effort qui a été fait pour bien rendre au niveau textuel les zones d'accélération (et donc d'allongement) de la veine fluide. Ainsi, si l'on peut encore travailler les animations superficielles des textures, le mouvement macroscopique lui est bien rendu dans les différentes zones d'écoulement (et donc les différents régimes) qui se succèdent. Par ailleurs, l'utilisateur peut contrôler bon nombres de paramètres, ce qui lui permet de définir le régime qu'il souhaite atteindre de façon assez simple. L'obtention de la forme visuelle souhaitée est donc relativement aisée et intuitive.

Dans la même optique de traitement quasi-2D, il est maintenant nécessaire de traiter le cas d'un obstacle émergent, c'est à dire d'une pierre dépassant au-dessus du niveau du ruisseau. Ce cas se différenciera du cas traité durant ce stage dans le sens où pour un obstacle émergent, la réduction de section de l'écoulement se fait au niveau de la largeur  $b$ , et non plus au niveau de la hauteur d'eau. C'est pourquoi on peut dire que ce problème est quasi-2D (presque 3D), contrairement au traitement purement 2D de notre seuil. Néanmoins, les approches sont voisines, et ce cas pourra à priori être traité en modélisant le problème sous la forme connue des piles de pont. Les ouvrages de référence cités en annexes contiennent d'ailleurs toute la bibliographie nécessaire au traitement de ce cas.

Une fois ces deux parties traitées, il sera intéressant de voir l'adaptation que l'on pourra en faire en 3D, sachant que le plus gros problème se posera pour les obstacles immergés. En effet pour des obstacles tels que des pierres, on n'aura pas une obstruction du ruisseau sur toute la largeur. Dès lors, il sera nécessaire de quantifier le débit et la zone réellement perturbés (probablement faudra-t-il adopter des coefficients de restriction "très empiriques"). Il faudra donc raffiner de façon importante le maillage aux abords des pierres, de façon à essayer de nous retrouver dans le cas traité ici, avec l'hypothèse que les particules fluides sont prisonnières du tube fluide auquel elle appartiennent à l'origine (conservation du débit dans un tube fluide). Des hypothèses fortes devront donc être prises, ce qui, en fonction du maillage et des échelles qui en découlent ne sera pas forcément évident. Cependant, il sera intéressant de voir ce que pourra donner un calcul avec une grille 3D bien adaptée et des hypothèses quantitatives quant aux relations entre débit, hauteur "de seuil" (ou plutôt dimensions des pierres) et hauteur d'eau.

Enfin, il sera utile de corréler les travaux déjà fait au niveau des ondes à la surface du fluide, de leurs perturbations, de leur entraînement par l'écoulement avec la modélisation précise des phénomènes générateurs : pierres immergées / émergées au sein du ruisseau. Ainsi, connaissant les paramètres de l'écoulement (dimension du ruisseau, débit, pente...) on devrait pouvoir déterminer la forme stationnaire de la surface libre du ruisseau au niveau des divers obstacles, et donc localiser les phénomènes générant des ondes de surface et peut-être même "quantifié" les ondes et perturbations elles-mêmes. Dès lors, l'animation "naturelle" de la surface du ruisseau devrait être prise en charge par la modélisation de

l'évolution dans le temps (au cours du transport) des ondes et perturbations engendrées par nos obstacles (sillages, tourbillons ...).

Comme on peut le voir, il reste encore pas mal de travail à fournir sur le sujet. Cela dit, la superposition des divers traitements particuliers à l'échelle d'un ruisseau semble pouvoir donner des résultats visuels intéressants et surtout nécessitant un coût de calcul raisonnable.

Quoi qu'il en soit il ne faut pas perdre de vue l'objectif qui est la synthèse d'images (temps réel ou semi-réel), ce qui encourage à mettre en place des méthodes générant des résultats visuels proches de la réalité sans pour autant nécessiter un temps de calcul aussi important que celui qu'impose la réelle simulation numérique en mécanique des fluides (qui elle, permet le calcul de tous les paramètres physiques en 3D avec une grande précision, mais bien évidemment pas l'animation temps réel ou quasi temps réel d'une scène).

Le résultat en synthèse d'image doit avant tout être visuellement correct, pas forcément numériquement précis, et surtout rapidement accessible et facilement contrôlable.

# Bibliographie

**[1]** Armando LENCASTRE, Hydraulique Générale - Editions Eyrolles 1983.

**[2]** Michel CARLIER, Hydraulique Générale et Appliquée - Editions Eyrolles 1972.

Chapitres utiles dans [1] et [2] : surface libre, écoulement par les déversoirs.

**[3]** Cours d'Écoulement des Fluides Réel de 2<sup>ème</sup> Année à l'ENSHMG :  
Écoulements à surface libre.

**[4]** Mason WOO, Jackie Neider, Tom DAVIS, OpenGL Programming Guide,  
Second Edition - Editions Addison-Wesley 1998.  
"Red Book"

## **ANNEXES : Code du Programme**