

FullSWOF_2D : un code pour le ruissellement

Olivier Delestre

MAPMO, Université d'Orléans

11 Décembre 2009



Position du problème



Lutte contre le ruissellement et l'érosion des sols

De l'amont...



(Photos : Yves Le Bissonais, INRA)

...à l'aval.



Aménagements au sein des zones amonts (bassins versants)



Aménagements au sein des zones amonts (bassins versants)



- ▶ D'où vient l'eau ?
- ▶ Où va-t-elle ?

Aménagements au sein des zones amonts (bassins versants)



- ▶ D'où vient l'eau ?
- ▶ Où va-t-elle ?

Utilisation de modèles pour :

- ▶ simuler les flux (localisation et volumes)
- ▶ prédire les chemins d'écoulement

Aménagements au sein des zones amonts (bassins versants)



- ▶ D'où vient l'eau ?
- ▶ Où va-t-elle ?

Utilisation de modèles pour :

- ▶ simuler les flux (localisation et volumes)
- ▶ prédire les chemins d'écoulement

~> proposer des aménagements

Modélisation et simulation numérique du ruissellement

Modélisation et simulation numérique du ruissellement

► Projet ANR METHODE



Les partenaires



Modélisation et simulation numérique du ruissellement

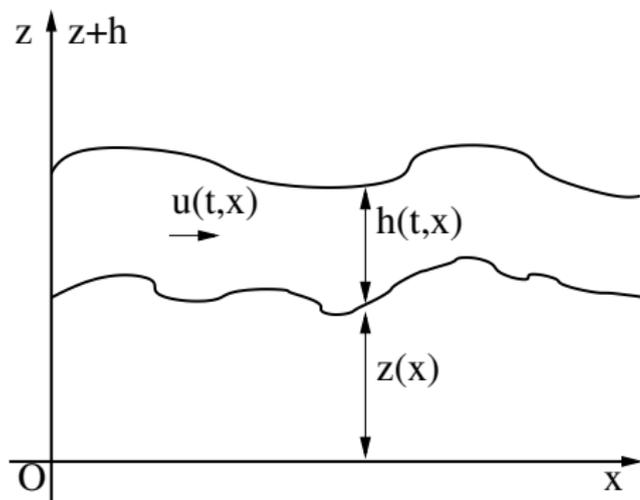
- ▶ Projet ANR METHODE



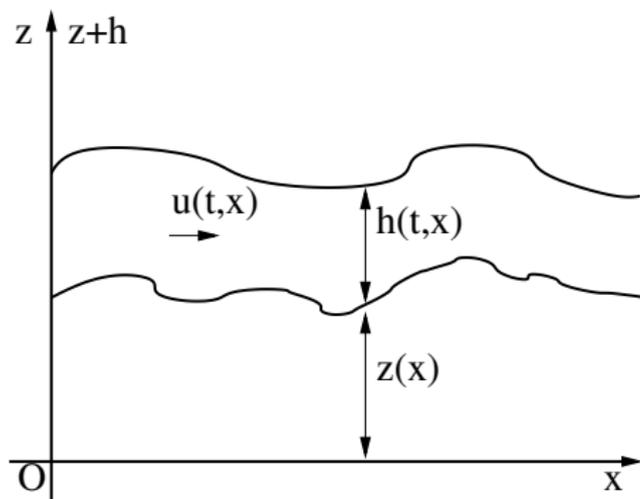
- ▶ ma thèse



Système de Saint-Venant



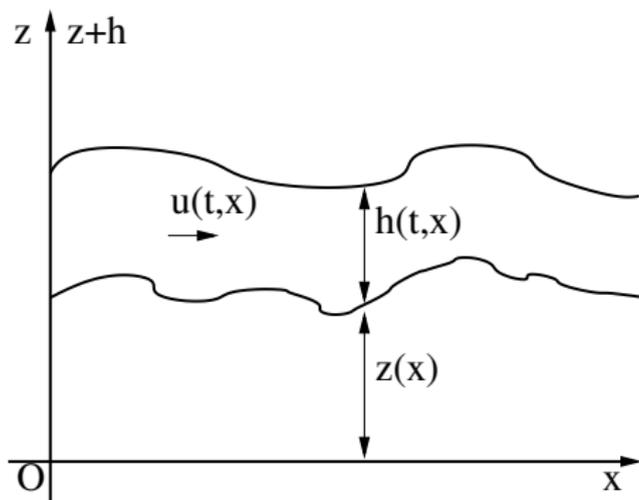
Système de Saint-Venant



Donnée : topographie z

Inconnues : vitesse u , hauteur d'eau h

Système de Saint-Venant



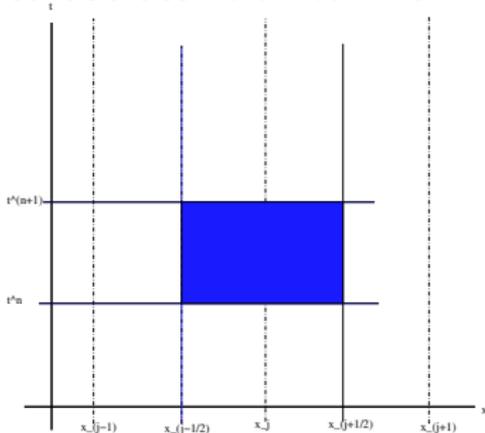
Donnée : topographie z

Inconnues : vitesse u , hauteur d'eau h

$$\begin{cases} \partial_t h + \partial_x(hu) = 0 \\ \partial_t(hu) + \partial_x(hu^2 + gh^2/2) = -gh\partial_x z \end{cases} \quad (1)$$

Calcul numérique des solutions

Méthode des volumes finis



On intègre

$$\partial_t U + \partial_x F(U) = 0$$

sur le volume

$$[t^n, t^{n+1}[\times]x_{j-1/2}, x_{j+1/2}[,$$

et on pose

$$U_j^n = \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} U(t^n, x) dx$$

Il vient

$$\Delta x \left[U_j^{n+1} - U_j^n \right] + \int_{t^n}^{t^{n+1}} F(U(t, x_{j+1/2})) dt - \int_{t^n}^{t^{n+1}} F(U(t, x_{j-1/2})) dt = 0$$

Il vient

$$\Delta x \left[U_j^{n+1} - U_j^n \right] + \int_{t^n}^{t^{n+1}} F(U(t, x_{j+1/2})) dt - \int_{t^n}^{t^{n+1}} F(U(t, x_{j-1/2})) dt = 0$$

On choisit une approximation des flux aux interfaces :

$$F_{j+1/2}^n \sim \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} F(U(t, x_{j+1/2})) dt,$$

Il vient

$$\Delta x \left[U_j^{n+1} - U_j^n \right] + \int_{t^n}^{t^{n+1}} F(U(t, x_{j+1/2})) dt - \int_{t^n}^{t^{n+1}} F(U(t, x_{j-1/2})) dt = 0$$

On choisit une approximation des flux aux interfaces :

$$F_{j+1/2}^n \sim \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} F(U(t, x_{j+1/2})) dt,$$

de sorte que la solution approchée (espérée...) sera donnée "de proche en proche" par

$$U_j^{n+1} = U_j^n + \frac{\Delta t}{\Delta x} [F_{j+1/2}^n - F_{j-1/2}^n]$$

- ▶ Chaque choix de $F_{j+1/2}^n$ détermine un schéma aux volumes finis :

HLL, cinétique, Rusanov, VFRoe-ncv, suliciu, ...

- ▶ Chaque choix de $F_{j+1/2}^n$ détermine un schéma aux volumes finis :

HLL, cinétique, Rusanov, VFRoe-ncv, suliciu, ...

- ▶ Ordre 2
 - ▶ en espace : MUSCL, ENO, ENO modifiée
 - ▶ en temps : Heun

- ▶ Chaque choix de $F_{j+1/2}^n$ détermine un schéma aux volumes finis :

HLL, cinétique, Rusanov, VFRoe-ncv, suliciu, ...

- ▶ Ordre 2
 - ▶ en espace : MUSCL, ENO, ENO modifiée
 - ▶ en temps : Heun
- ▶ Coupler avec le second membre (topographie $\partial_x z$)
Nécessité : compatibilité avec les états d'équilibre

Etats d'équilibre



Etats d'équilibre

$$\begin{cases} \partial_t h + \partial_x(hu) = 0 \\ \partial_t(hu) + \partial_x(hu^2 + gh^2/2) = -gh\partial_x z \end{cases} \quad (2)$$

Etats d'équilibre

$$\begin{cases} \partial_t h + \partial_x(hu) = 0 \\ \partial_t(hu) + \partial_x(hu^2 + gh^2/2) = -gh\partial_x z \end{cases} \quad (2)$$

$$\partial_t h = \partial_t u = \partial_t q = 0$$

Etats d'équilibre

$$\begin{cases} \partial_t h + \partial_x(hu) = 0 \\ \partial_t(hu) + \partial_x(hu^2 + gh^2/2) = -gh\partial_x z \end{cases} \quad (2)$$

$$\partial_t h = \partial_t u = \partial_t q = 0$$

$$\begin{cases} hu = Cte \\ u^2/2 + g(h+z) = Cte \end{cases} \cdot$$

Etats d'équilibre

$$\begin{cases} \partial_t h + \partial_x(hu) = 0 \\ \partial_t(hu) + \partial_x(hu^2 + gh^2/2) = -gh\partial_x z \end{cases} \quad (2)$$

$$\partial_t h = \partial_t u = \partial_t q = 0$$

$$\begin{cases} hu = Cte \\ u^2/2 + g(h+z) = Cte \end{cases} \cdot$$

On considère

$$\begin{cases} u = Cte \\ g(h+z) = Cte \end{cases} \cdot$$

Reconstruction hydrostatique [Audusse et al., 2004] et
[Bouchut, 2004]

On définit

$$z^* = \max(z_G, z_D)$$

et

$$\begin{cases} U_G^* = (h_G^*, h_G^* u_G), & U_D^* = (h_D^*, h_D^* u_D) \\ h_G^* = \max(h_G + z_G - z^*, 0) \\ h_D^* = \max(h_D + z_D - z^*, 0) \end{cases} .$$

Frottements et pluie

Système de Saint-Venant avec une **force extérieure** f

$$\begin{cases} \partial_t h + \partial_x(hu) = R, \\ \partial_t(hu) + \partial_x(hu^2 + gh^2/2) + h\partial_x z = hf, \end{cases} \quad (3)$$

R intensité de pluie

$f = f(h, u)$ force de frottement (sur le fond)

Divers modèles possibles

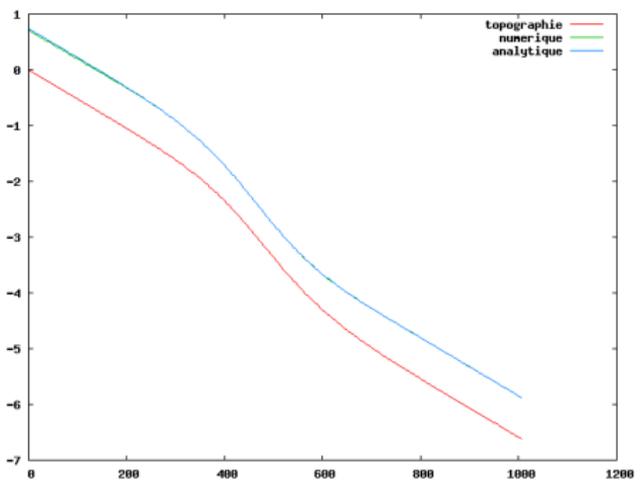
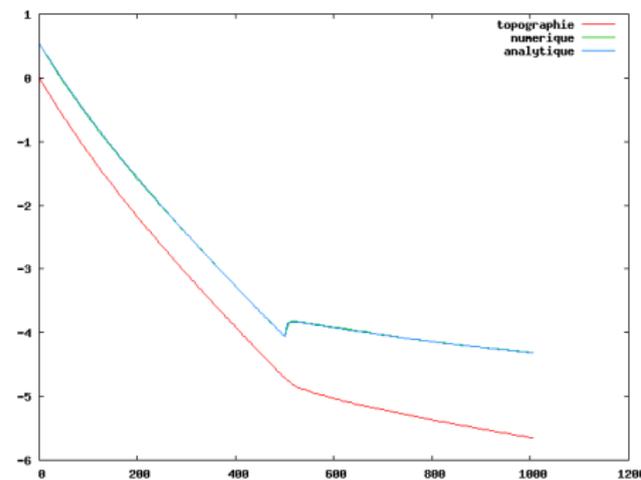
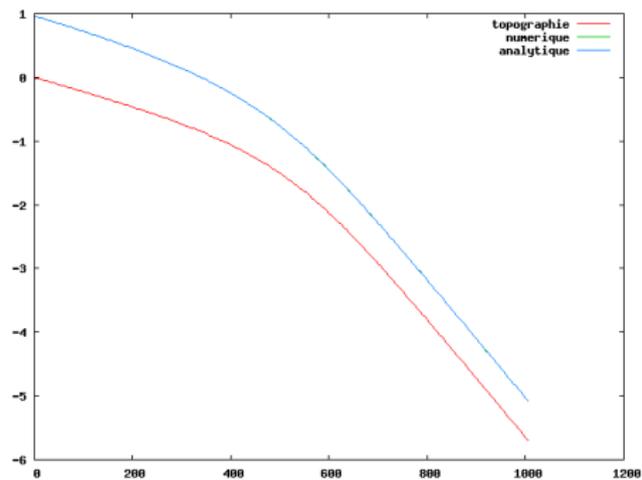
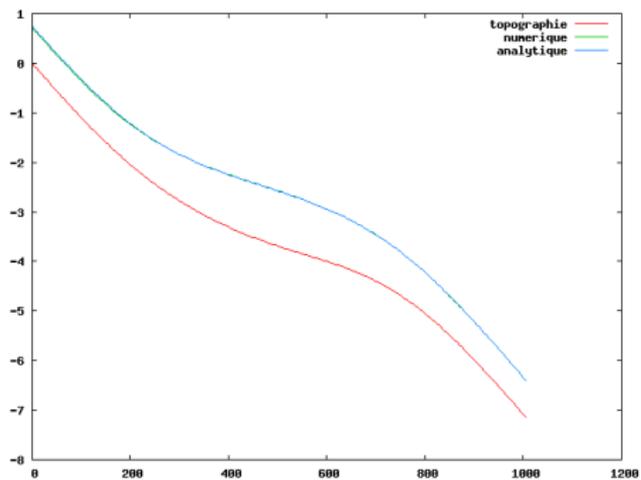
► Manning : $f = n^2 \frac{u|u|}{h^{4/3}}$

► Darcy-Weisbach : $f = F \frac{u|u|}{8gh}$

Test 1 : Solutions analytiques (Vo Thi Ngoc, M2-MAPMO)

Cas test originaux :

- ▶ Saint-Venant :
 - ▶ donnée z
 - ▶ inconnues h et u (et donc q)
- ▶ cas test
 - ▶ données h et q (et donc u)
 - ▶ inconnue z

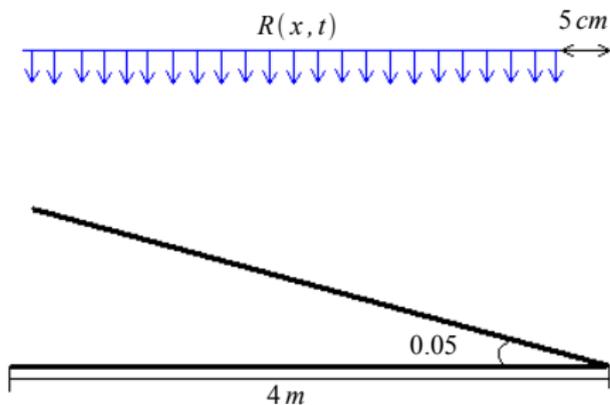


Test 2 : Expérimental (INRA) & analytique

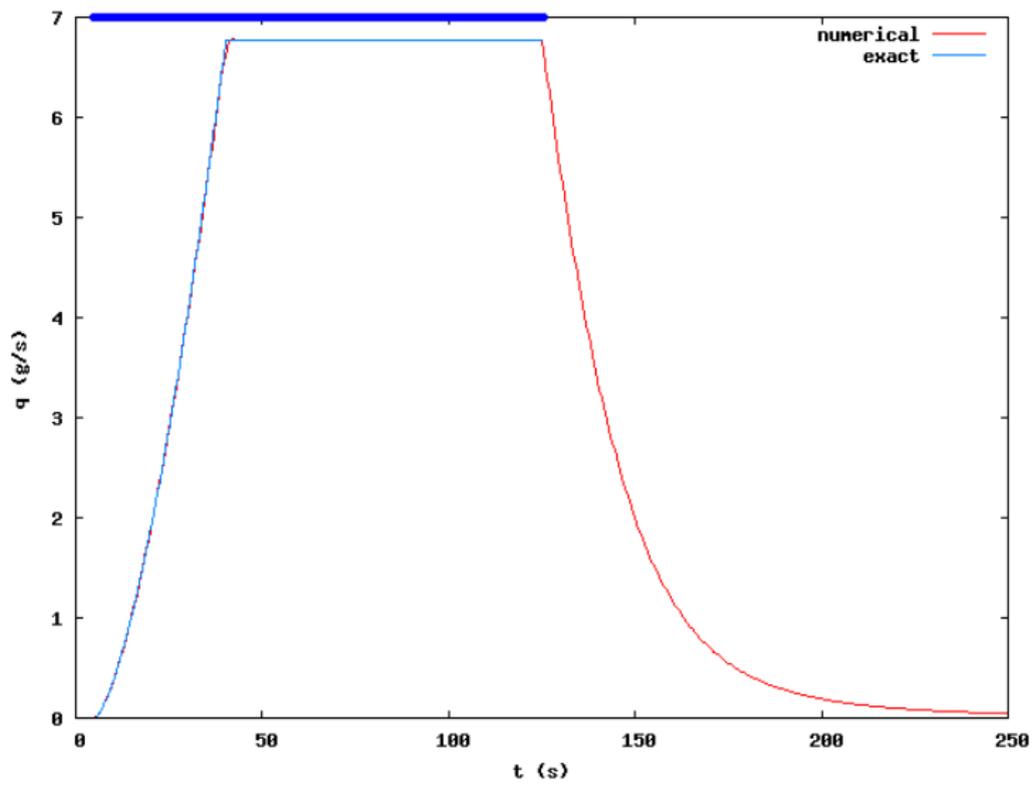


Données

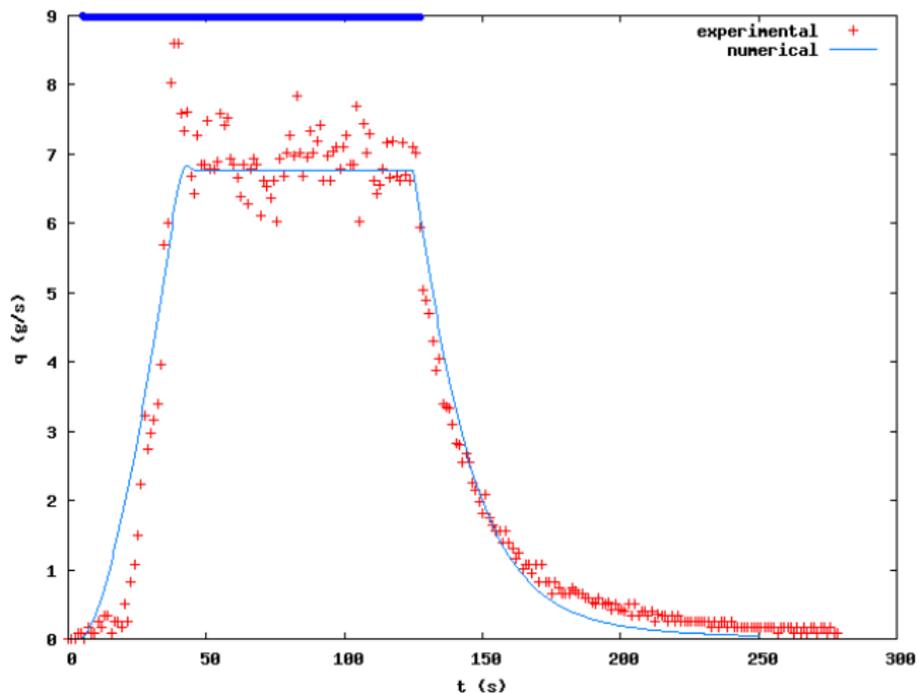
$$0 \leq t \leq 250\text{s}$$



$$R(x, t) = \begin{cases} 50 \text{ mm/h} & \text{si } (x, t) \in [0, 3.95] \times [5, 125] \\ 0 & \text{sinon} \end{cases}$$



Un résultat de simulation



Frottement de Darcy-Weisbach $F \frac{u|u|}{8gh}$
Caler le coefficient F .

Avant FullSWOF_2D

- ▶ fortran 1d

- ▶ fortran 2d

BRGM : EL Bouajaji (M2,2007), Rousseau (M2,2008)

Avant FullSWOF_2D

- ▶ fortran 1d

- ▶ fortran 2d

BRGM : EL Bouajaji (M2,2007), Rousseau (M2,2008)
problèmes de modularité et **tests redondants**

Avant FullSWOF_2D

- ▶ fortran 1d
- ▶ fortran 2d
BRGM : EL Bouajaji (M2,2007), Rousseau (M2,2008)
problèmes de modularité et **tests redondants**

FullSWOF_2D (Full Shallow Water for Overland Flow)

- ▶ C++ 2d
INRA : Delestre (M2,2008)

Développement : Delestre, Laguerre, Razafison, Rousseau

Modularité

<i>flux</i>
<ul style="list-style-type: none">- <i>f1</i>- <i>f2</i>- <i>tx</i>
<ul style="list-style-type: none">+ <i>calcul(double, double, double, double) : void</i>+ <i>set_tx(double) : void</i>+ <i>get_f1() : double</i>+ <i>get_f2() : double</i>+ <i>get_cfl() : double</i>
<i>flux()</i>

Ordre 1

Initialisation des variables

for $t = 0$ to N **do**

Reconstruction hydrostatique

Calcul des flux numériques : HLL, Rusanov, Cinétique, ...

Application du schéma

Traitement des frottements

end for

Ordre 2

Initialisation des variables

for $t = 0$ to N **do**

Reconstruction ordre 2 en x : MUSCL, ENO ou ENO modifié

Reconstruction hydrostatique

Calcul des flux numériques : HLL, Rusanov, Cinétique, ...

Application du schéma

Traitement des frottements

Reconstruction ordre 2 en x : MUSCL, ENO ou ENO modifié

Reconstruction hydrostatique

Calcul des flux numériques : HLL, Rusanov, Cinétique, ...

Application du schéma

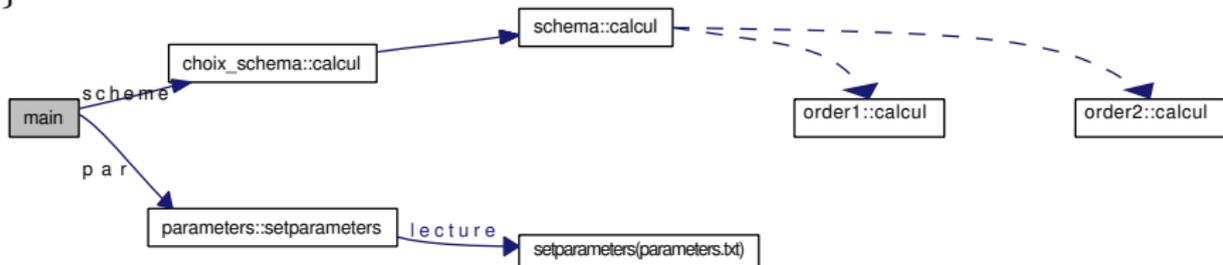
Traitement des frottements

Ordre 2 en t : Méthode de Heun

end for

FullSWOF_2D.cpp

```
int main(int argc, char ** argv) {  
    parameters par ;  
    par.setparameters(" parametres.txt" );  
    choix_schema * schema ;  
    schema = new choix_schema(par) ;  
    cout << "schema choisi !" << endl ;  
    schema->calcul() ;  
    cout << "calcul ok !" << endl ;  
    delete schema ;  
    return 0 ;  
}
```



Choix_schema.cpp

```
#include "choix_schema.h"
```

```
#include <divers.h>
```

```
choix_schema : :choix_schema(parameters & par){
```

```
    switch (par.get_ordre()){
```

```
        case 1 :
```

```
            Sche = new ordre1(par);
```

```
            break;
```

```
        case 2 :
```

```
            Sche = new ordre2(par);
```

```
            break;
```

```
        default :
```

```
            int k=3;
```

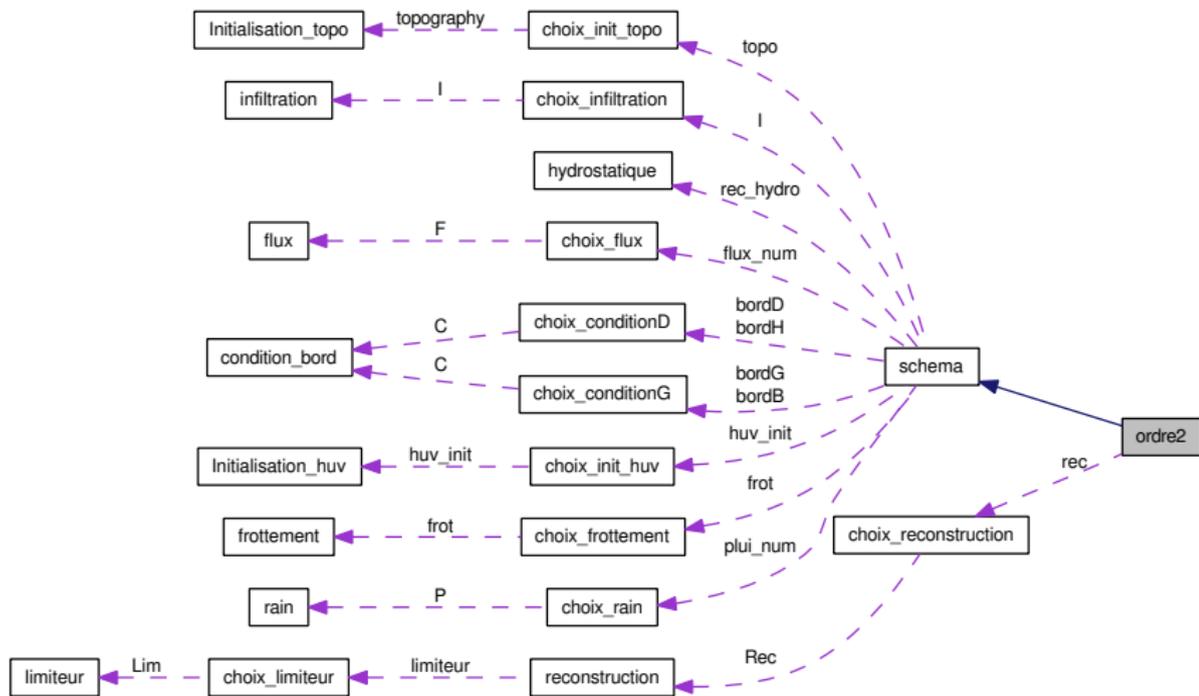
```
    }
```

```
}
```

```
void choix_schema : :calcul(){ Sche->calcul(); }
```

```
choix_schema : : choix_schema(){ delete Sche; }
```

Ordre2



Documentation

```
/*
*****
***** F_Rusanov.h *****
*****
*
* A class of FullSWOF_2D
* Done by Olivier Delestre, october 2008, olivierdelestre41@yahoo.fr
* Modified by ..., ..., ...@...
* Copyright (C) 2008 Olivier Delestre
*
*****/

#include <divers.h>

#ifdef Classe_flux
#include "flux.h"
#endif

#ifdef Classe_F_Rusanov
#define Classe_F_Rusanov

/*!
* \file F_Rusanov.h
* \brief Rusanov flux
* \author Olivier Delestre
* \version 1.0.0
*/

/*! \class F_Rusanov
* \brief class which allows to calculate Rusanov flux
*
* \f[{\cal F}(U_G,U_D)=\frac{F(U_G)+F(U_D)}{2}-c.\frac{U_D-U_G}{2},\f]
with
\f[ c=\max(|u_G|+\sqrt{g h_G},|u_D|+\sqrt{g h_D}) .\f]
*/

class F_Rusanov: public flux{
```

Doxygen

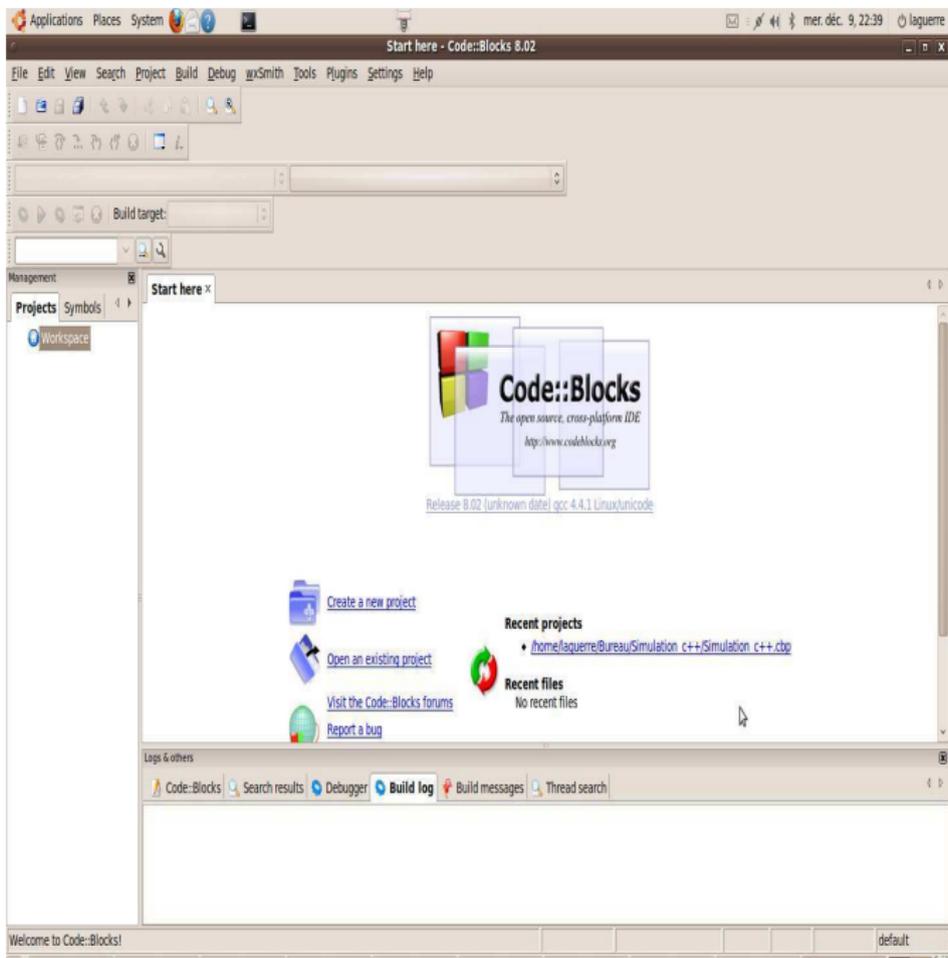
The screenshot shows a Mozilla Firefox browser window displaying the class reference for `F_Rusanov`. The browser's address bar shows the file path `file:///home/olivier/these/doxygen3/FullSWOF_2D/doc/html/index.html`. The page title is `FullSWOF_2D`. The left sidebar contains a navigation menu with options: `Class List`, `Class Hierarchy`, `Class Members`, `Graphical Class Hierarchy`, and `File List`. The main content area is titled `F_Rusanov Class Reference` and includes the following information:

- Description: class which allows to calculate Pusanov flux More...
- Code: `#include <F_Rusanov.h>`
- Inherits: `flux`
- Collaboration diagram for `F_Rusanov`:

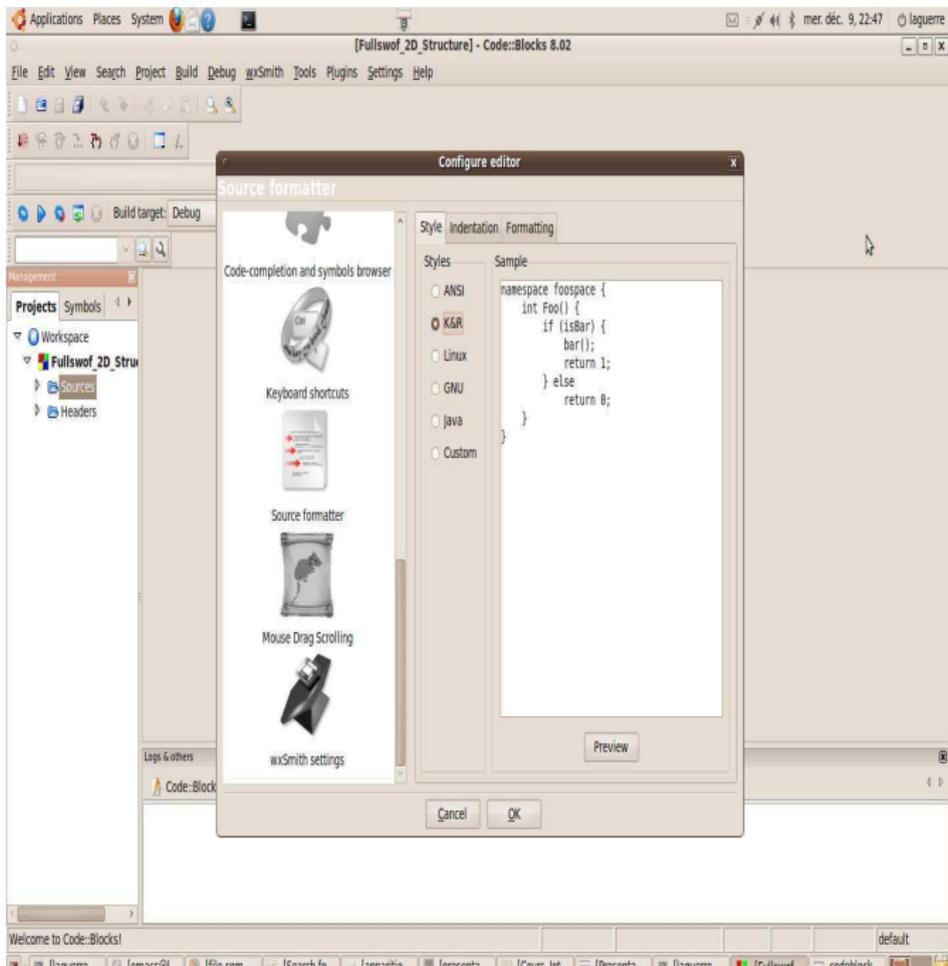
```
graph BT; F_Rusanov --> flux;
```
- List of all members:
- Public Member Functions:
 - `F_Rusanov()` Constructor.
 - `void calc14 (PEEL, PEEL, PEEL, PEEL, PEEL, PEEL)` Calculates Pusanov flux.
 - `virtual ~F_Rusanov()` Destructor.
- Detailed Description: class which allows to calculate Pusanov flux
- Equation:
$$\mathcal{F}(U_G, U_D) = \frac{\mathcal{F}(U_G) + \mathcal{F}(U_D)}{2} - c \frac{U_D - U_G}{2}$$
- with

The browser's taskbar at the bottom shows the text "Terminé" and the active window "Figures - Navigateur de fichiers".

Code : :Blocks



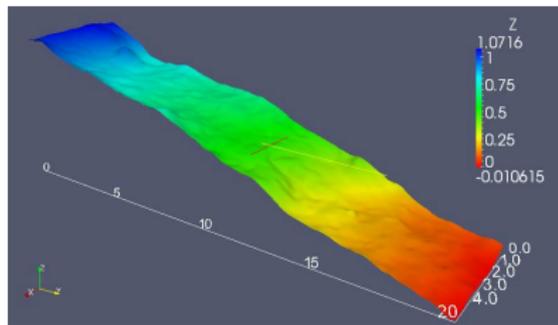
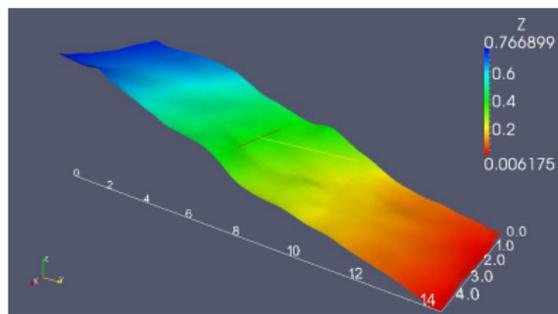
Code : :Blocks



Parcelles au Niger ([Esteves et al., 2000], IRD)

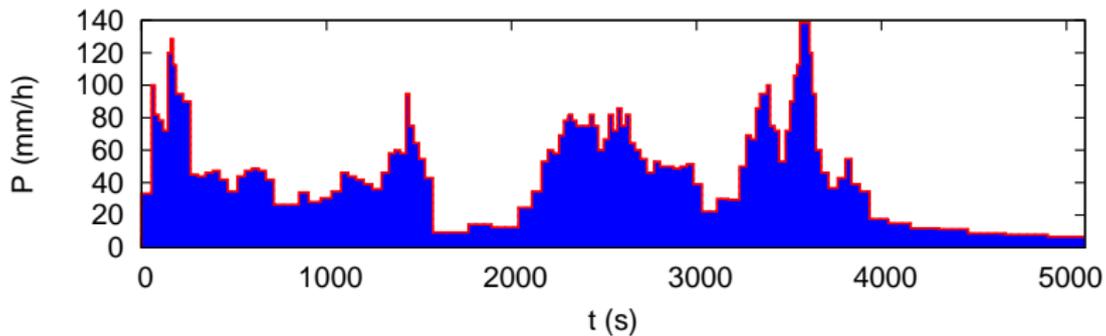


Parcelles au Niger ([Esteves et al., 2000], IRD)

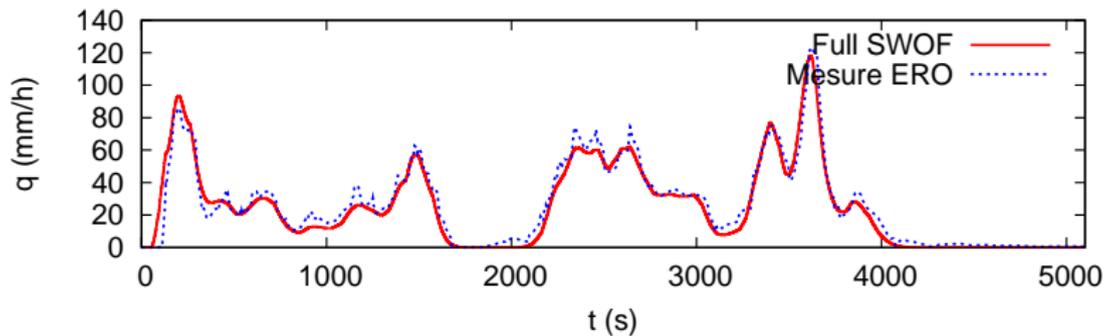


Darcy-Weisbach

Hyetogramme

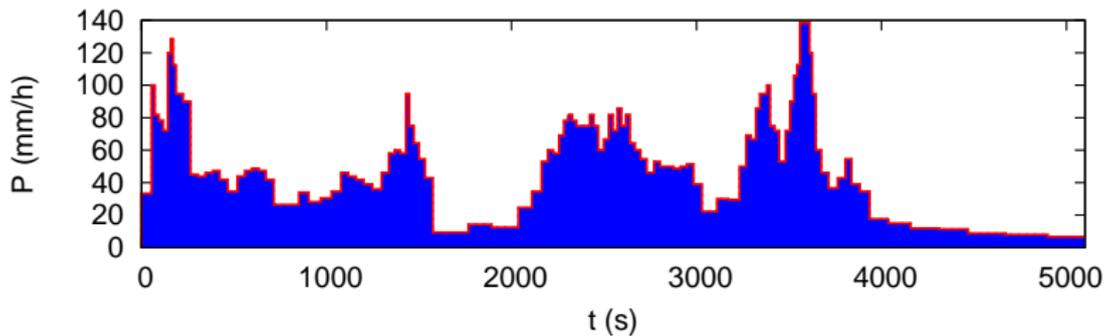


Hydrogramme

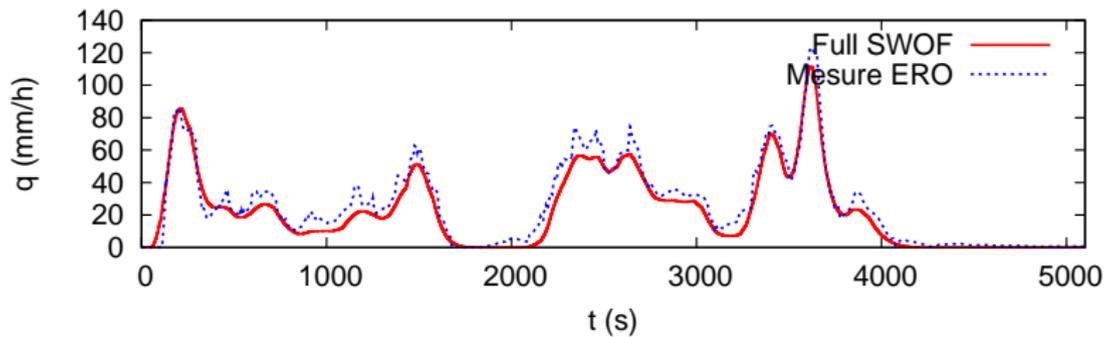


Manning

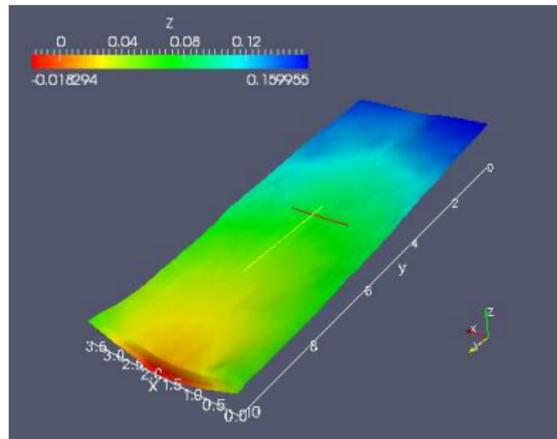
Hyetogramme



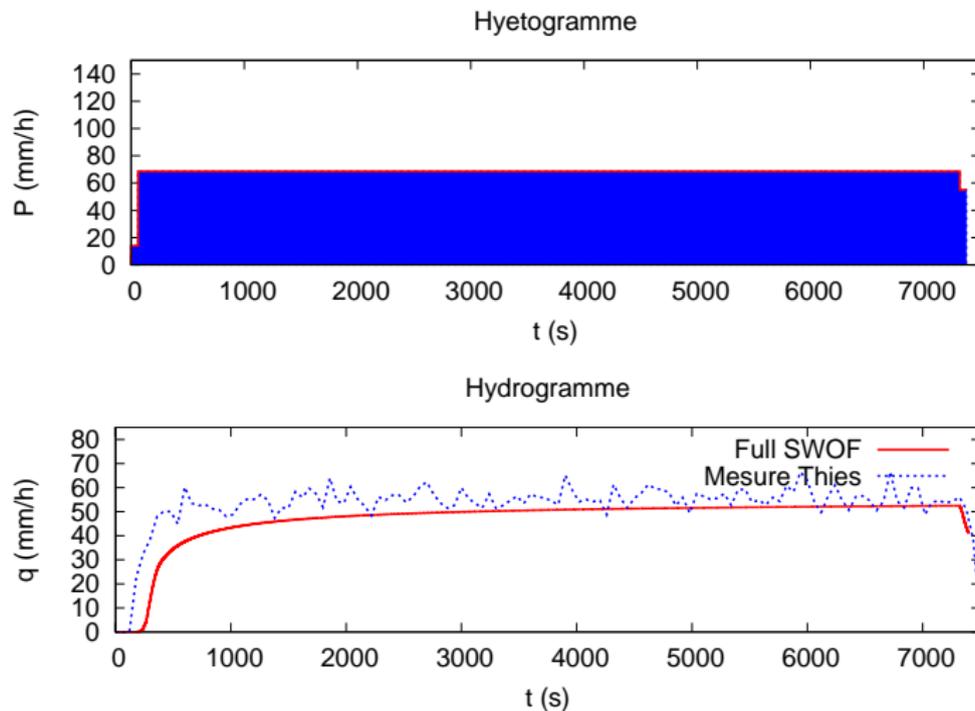
Hydrogramme



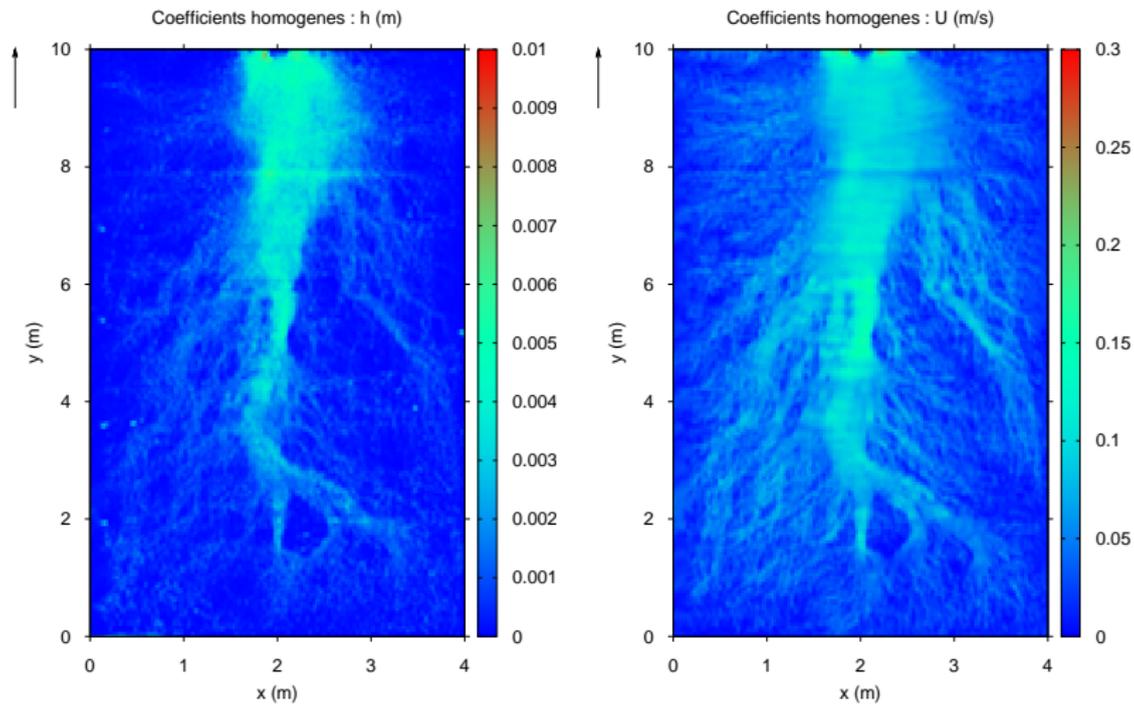
Parcelle de Thies au Sénégal ([Tatard et al., 2008], IRD)



Parcelle de Thies au Sénégal ([Tatard et al., 2008], IRD)



Parcelle de Thies au Sénégal ([Tatard et al., 2008], IRD)





Audusse, E., Bouchut, F., Bristeau, M.-O., Klein, R., and Perthame, B. (2004).

A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows.

SIAM J. Sci. Comput., 25(6) :2050–2065.



Bouchut, F. (2004).

Nonlinear stability of finite volume methods for hyperbolic conservation laws, and well-balanced schemes for sources, volume 2/2004.

Birkhäuser Basel.



Esteves, M., Faucher, X., Galle, S., and Vauclin, M. (2000).

Overland flow and infiltration modelling for small plots during unsteady rain : numerical results versus observed values.

Journal of Hydrology, 228 :265–282.



Tatard, L., Planchon, O., Wainwright, J., Nord, G., Favis-Mortlock, D., Silvera, N., Ribolzi, O., Esteves, M., and Huang, C.-h. (2008).

Measurement and modelling of high-resolution flow-velocity data under simulated rainfall on a low-slope sandy soil.

Journal of Hydrology, 348(1-2) :1–12.

flux HLL

$$\mathcal{F}(U_G, U_D) = \begin{cases} F(U_G) & \text{si } 0 < c_1 \\ F(U_D) & \text{si } c_2 < 0 \\ \frac{c_1 F(U_G) - c_2 F(U_D)}{c_2 - c_1} + \frac{c_1 c_2 (U_D - U_G)}{c_2 - c_1} & \text{sinon} \end{cases},$$

avec deux paramètres

$$c_1 < c_2.$$

flux HLL

$$\mathcal{F}(U_G, U_D) = \begin{cases} F(U_G) & \text{si } 0 < c_1 \\ F(U_D) & \text{si } c_2 < 0 \\ \frac{c_1 F(U_G) - c_2 F(U_D)}{c_2 - c_1} + \frac{c_1 c_2 (U_D - U_G)}{c_2 - c_1} & \text{sinon} \end{cases},$$

avec deux paramètres

$$c_1 < c_2.$$

Pour c_1 et c_2 , on prend

$$c_1 = \inf_{U=U_G, U_D} \left(\inf_{j \in \{1,2\}} |\lambda_j(U)| \right) \text{ and } c_2 = \sup_{U=U_G, U_D} \left(\sup_{i \in \{1,2\}} |\lambda_i(U)| \right).$$

avec $\lambda_1(U) = u - \sqrt{gh}$ et $\lambda_2(U) = u + \sqrt{gh}$. [▶ retour](#)