VOLUMES FINIS
PROJET 4

PIAO Chenghe
SONZOGNI Max

2018
Polytech Montpellier

I) Analyse mathématique

On va simuler l'écoulement d'eau de faible profondeur avec un fond plat en une dimension. L'eau est considérée comme un fluide incompressible non visqueux et les frottements air/eau et eau/sol seront négligés.

Les inconnues, fonction du temps $t \in \mathbb{R}^+$ et de l'espace $x \in \mathbb{R}$, sont h la hauteur d'eau et u la vitesse horizontale. Soit $g > 0$ la constante de gravité, on a le système suivant :

$$h_t + (hu)_x = 0$$

$$(hu)_t + \left(hu^2 + \frac{gh^2}{2}\right)_x = 0$$

Sous forme vectorielle, on a

$$U_t + \left(F(U)\right)_x = 0$$

avec

$$U = \begin{pmatrix} h \\ hu \end{pmatrix}, \quad F(U) = \begin{pmatrix} hu \\ hu^2 + gh^2/2 \end{pmatrix}$$

Si on développe ce système, on obtient

$$\begin{cases} h_t + h_x u + hu_x = 0 \\ h_t u + hu_t + h_x u^2 + 2uu_x h + ghh_x = 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} h_t = -(h_x u + hu_x) \\ u_t = ((h_x u + hu_x)u - h_x u^2 - 2uu_x h - ghh_x)/h \end{cases}$$

$$\Leftrightarrow \begin{cases} h_t + h_x u + hu_x = 0 \\ u_t + gh_x + uu_x = 0 \end{cases}$$

On peut alors réécrire le système sous la forme

$$W_t + A(W)W_x = 0$$

avec

$$W = \begin{pmatrix} h \\ u \end{pmatrix}, \quad A(W) = \begin{pmatrix} u & h \\ g & u \end{pmatrix}$$

Pour déterminer la condition de CFL que nous allons utiliser pour modéliser numériquement ce problème, nous avons besoin des valeurs propres de la matrice *A(W)*.

Ainsi, on a :

$$det(A(W) - \lambda.Id) = \begin{bmatrix} u-\lambda & h \\ g & u-\lambda \end{bmatrix} = \lambda^2 - 2u\lambda + u^2 - gh$$

On cherche alors λ tels que $det(A(W)-\lambda.Id)=0$, ce qui donne

$$\lambda_1 = u+\sqrt{gh}\ , \qquad \lambda_2 = u-\sqrt{gh}$$

$$A(W).r_k = \lambda_k r_k$$

$$\Leftrightarrow \begin{cases} ua_k + hb_k = \lambda_k a_k \\ ga_k + ub_k = \lambda_k b_k \end{cases}$$

On a donc

$$\begin{cases} a_1 = \sqrt{\dfrac{h}{g}}b_1 \\ \\ a_2 = -\sqrt{\dfrac{h}{g}}b_2 \end{cases}$$

On a alors comme vecteurs propres

$$r_1 = \left(\sqrt{\frac{h}{h+g}}\ ;\ \sqrt{\frac{g}{h+g}}\right)\ , \qquad r_2 = \left(\sqrt{\frac{h}{h+g}}\ ;\ -\sqrt{\frac{g}{h+g}}\right)$$

Finalement, on obtient

$$\begin{cases} (\nabla\lambda_1)^t.r_1 = \dfrac{3}{2}\sqrt{\dfrac{g}{h}} \\ (\nabla\lambda_2)^t.r_2 = -\dfrac{3}{2}\sqrt{\dfrac{g}{h}} \end{cases}$$

On pose les fonctions $\eta(W) = \frac{hu^2}{2} + \frac{gh^2}{2}$ et $\phi(W) = \frac{hu^3}{2} + gh^2u$

On a alors

$$\nabla_W\phi(W) = \left(\frac{u^3}{2} + 2ghu\ ;\ \frac{3}{2}hu^2 + gh^2\right)$$

et

$$\nabla_W\eta(W).A(W) = \left(\frac{u^2}{2} + gh\ ; hu\right).\begin{pmatrix} u & h \\ g & u \end{pmatrix} = \left(\frac{u^3}{2} + 2ghu\ ;\ \frac{3}{2}hu^2 + gh^2\right)$$

On a donc $\nabla_W\phi(W) = \nabla_W\eta(W).A(W)$, ce qui démontre que $\eta(W)$ est une entropie du système, avec comme flux d'entropie $\phi(W)$.

II) Modélisation numérique

On s'intéresse maintenant au problème de Riemann $U(0,x) = \begin{cases} U_L & si\ x < 0 \\ U_R & si\ x > 0 \end{cases}$

On considère le schéma volumes finis de Rusanov :

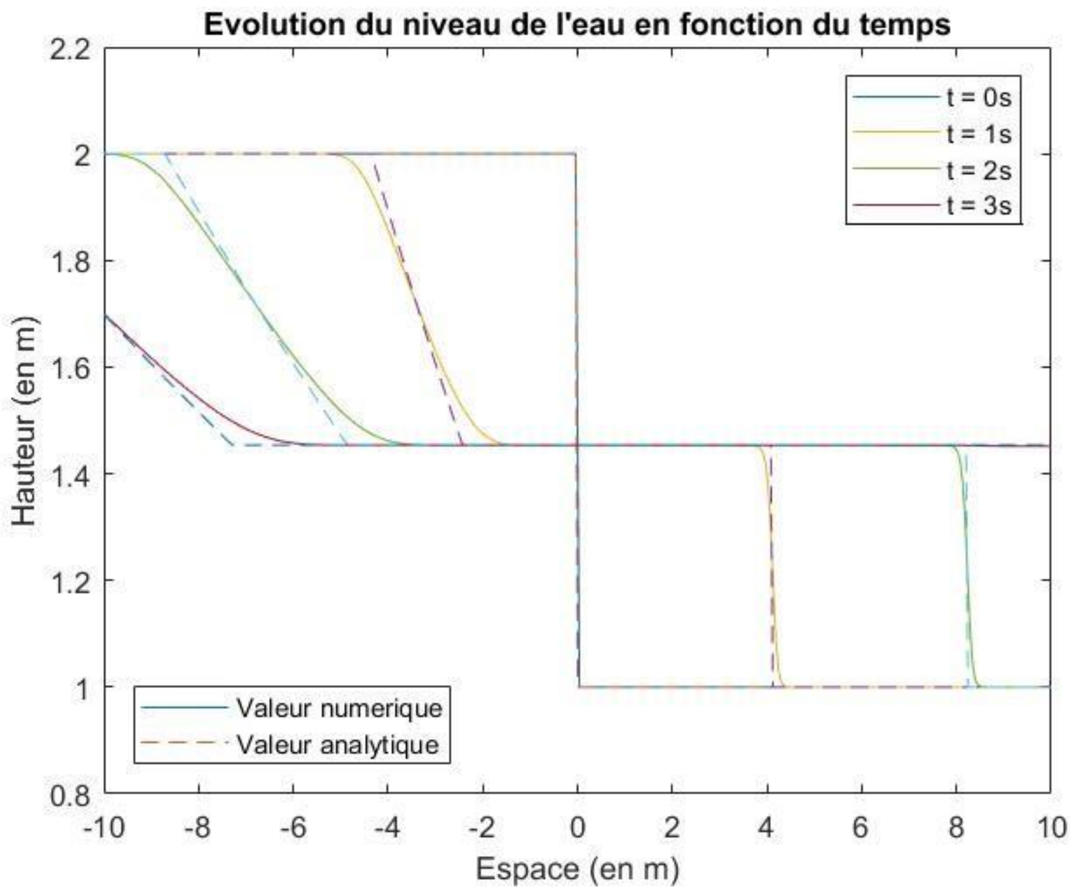
$$U_i^{n+1} = U_i^n - \frac{\delta t}{\delta x}\left(G(U_i^n, U_{i+1}^n) - G(U_{i-1}^n, U_i^n)\right)$$

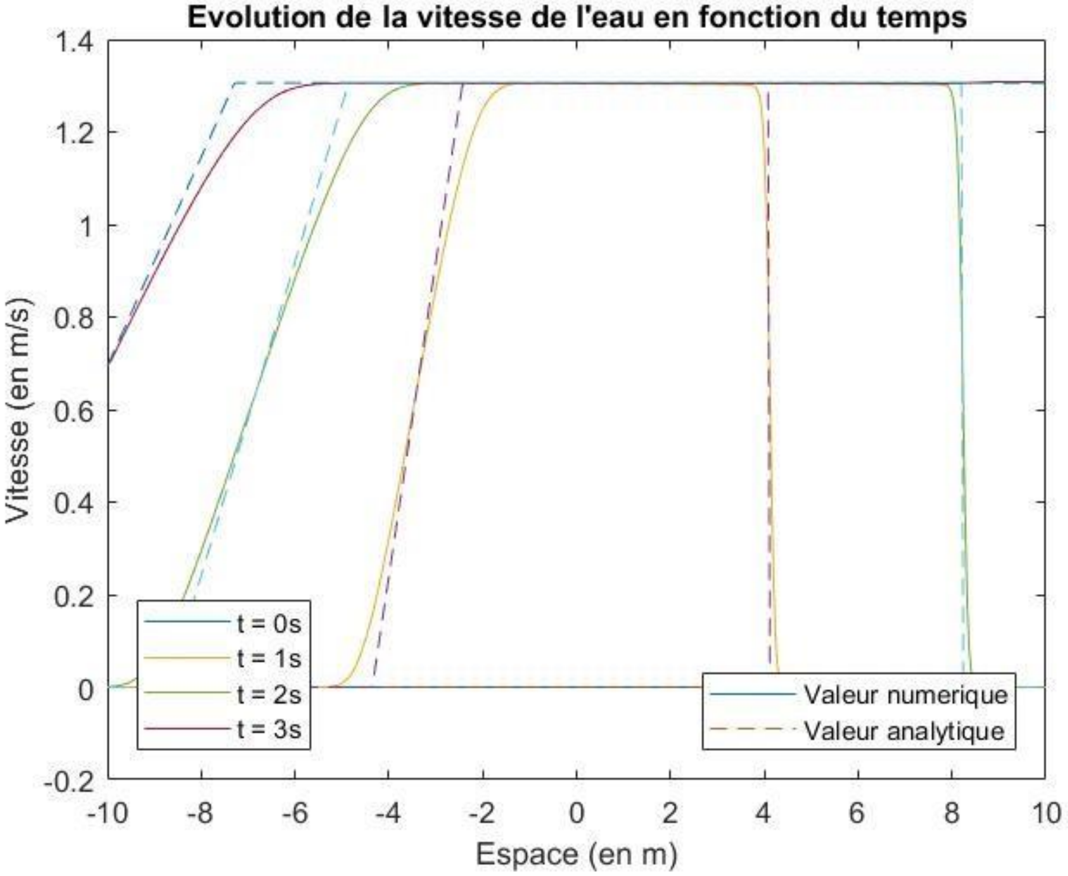
avec

$$G(u,v) = \frac{F(u) + F(v)}{2} - max_{k=1,2}(|\lambda_k(u)|, |\lambda_k(v)|)\frac{v-u}{2}$$

La condition CFL est $2max_i(|\lambda_1(U_i^n)|, |\lambda_2(U_i^n)|)\delta t \leq \delta x$

On considère un bassin [-10, 10], séparé en 2 parties égales grâce à une paroi située en x=0. A t<0, la paroi gauche est remplie avec de l'eau immobile à une hauteur h_L = 2 et à droite h_R = 1. On retire la paroi à l'instant t=0. On obtient alors les évolutions d'hauteur et de vitesse de l'eau suivantes :





On peut tout d'abord remarquer que la solution numérique est très proche de la solution analytique (voir annexes pour les formules), ce qui permet de dire que la modélisation constitue une bonne approximation de ce problème, et peut donc être utilisée pour d'autres problèmes similaires.

On voit ensuite que l'eau passe de la partie gauche à la partie droite en effectuant un ''plateau'' à un niveau légèrement inférieur à la moyenne entre les niveaux haut et bas, tandis que c'est à cet endroit que sa vitesse est la plus élevée. Une fois le bord gauche atteint par le plateau, le niveau de l'eau se stabilise à la hauteur du plateau, et sa vitesse devient constante le long du bassin.

Pour estimer la valeur exacte de la solution de ce problème, il a fallu utiliser une méthode de calcul implicite. Nous en avons utilisé 2 pour modéliser notre problème : la méthode de Newton et la méthode de la sécante. La première est plus/moins rapide, avec un temps de calcul du programme d'environ 12.27 secondes, tandis que la seconde a un temps de calcul tournant autour de 12.30 secondes. Pour ce problème, c'est donc la méthode de Newton qui est la plus efficace.

Annexe 1 : Programme principal

```
g = 9.81;

x0 = -10;
xf = 10;
Nx = 500;
dx = (xf-x0)/Nx;
x = [x0:dx:xf];
xtilde = [x0-(dx/2):dx:xf+(dx/2)];

t0 = 0;
tf = 3;

%definition des valeurs propres
lbd1 = @(u,h)(u+sqrt(g*h));
lbd2 = @(u,h)(u-sqrt(g*h));

%creation de h et de u
h = zeros(length(x),1);
u = h;
for xi=1:length(x)
        h(xi,1) = (1/dx)*quadv(@f,xtilde(xi),xtilde(xi+1)); %f definie dans le fichier f.m
end

t = 0; n = 0;
while t<tf
        n = n+1;
        dt = max((dx/2)/max(abs(lbd1(u(:,n),h(:,n))),abs(lbd2(u(:,n),h(:,n))))); %calcul de dt
        t = t+dt;

        U = @(i,n)([h(i,n)
           h(i,n).*u(i,n)]);
        maxLbd =
@(u,v)(max(max(abs(lbd1(u(2)/u(1),u(1))),abs(lbd1(v(2)/v(1),v(1)))),max(abs(lbd2(u(2)/u(1),u(1))),abs(lbd2(
v(2)/v(1),v(1))))));
        F = @(u)([u(2)
            (u(2).^2)/u(1)+g/2*u(1).^2]);
        G = @(u,v)(0.5*(F(u)+F(v)-(v-u)*maxLbd(u,v)));

        for i=2:length(x)-1
                P = U(i,n)-(dt/dx).*(G(U(i,n),U(i+1,n))-G(U(i-1,n),U(i,n)));
                h(i,n+1) = P(1);
                u(i,n+1) = P(2)/P(1);
        end
        %conditions aux limites : comme la vitesse et le niveau de leau sont constants sur les bords, on a
une derivee nulle
        h(1,n+1) = h(2,n+1); u(1,n+1) = u(2,n+1);
        h(length(x),n+1) = h(length(x)-1,n+1); u(length(x),n+1) = u(length(x)-1,n+1);

        Wexxt = Wex(t,x); %calcul de la solution exacte (voir Wex.m)

        %affichage des solutions
        plot(x,h(:,n),'r.',x,Wexxt(1,:),'r',x,u(:,n),'b.',x,Wexxt(2,:),'b')
        legend({'Hauteur calculee','Hauteur exacte','Vitesse calculee','Vitesse exacte'})
        pause(0.001)
end
```

Annexe 2 : fonction f

```
function f = f(x)

hl = 2;
hr = 1;

for i=1:length(x)
	if x(i)<0
		f(i) = hl;
	else
		f(i) = hr;
	 end
end

end
```

<u>Annexe 3</u> : fonction Wex

```
function Wex = Wex(t,x)

ul = 0; ur = 0;
hl = 2; hr = 1;
g = 9.81;
meth = 1 ;

F = @(h)(ul-ur+2*sqrt(g)*(sqrt(hl)-sqrt(h))+(hr-h)*sqrt((g*(h+hr))/(2*h*hr)));
Fprime = @(h)(-sqrt(g/h)-sqrt((g*(h+hr))/(2*h*hr))+0.5*(hr-h)*sqrt((2*h*hr)/(g*(h+hr))));

If meth == 1 %methode de la secante
        emax = 0.01; imax = 10; i = 0;
        htoile1 = hl; htoile2 = hr;
        while i<imax && abs(F(htoile2))>emax
                htoile = htoile2 - (htoile2-htoile1)*F(htoile2)/(F(htoile2)-F(htoile1));
                htoile1 = htoile2; htoile2 = htoile;
                i = i+1;
        end

else %methode de Newton
        emax = 0.01; imax = 10; i = 0;
        htoile = hl;
        while i<imax && abs(F(htoile))>emax
                htoile = htoile - F(htoile)/Fprime(htoile);
                i = i+1;
        end
end

utoile = ul+2*sqrt(g)*(sqrt(hl)-sqrt(htoile));

Wex = zeros(2,length(x));
for j=1:length(x)
         if x(j) < (ul-sqrt(g*hl)*t) %cas 1
                Wex(:,j) = [hl
                   ul];
        elseif x(j) > (ul-sqrt(g*hl)*t) && x(j) < (utoile-sqrt(g*htoile))*t %cas 2
                Wex(:,j) = [((ul+2*sqrt(g*hl)-x(j)/t).^2)/(9*g)
                   (ul+2*sqrt(g*hl)+2*x(j)/t)/3];
        elseif x(j) > (utoile-sqrt(g*htoile))*t && x(j) < (utoile + hr*sqrt((g*(htoile+hr))/(2*htoile*hr)))*t %cas 3
                Wex(:,j) = [htoile
                   utoile];
        else %dernier cas
                Wex(:,j) = [hr
                   ur];
        end
end

end
```

Annexe 4 : solution exacte (analytique)

La solution exacte de ce problème s'écrit :

$$W(t,x) = \begin{cases} W_L & si\ x < \left(u_L - \sqrt{gh_L}\right)t \\ W_1 & si\ \left(u_L - \sqrt{gh_L}\right)t < x < \left(u^* - \sqrt{gh^*}\right)t \\ W^* & si\ \left(u^* - \sqrt{gh^*}\right)t < x < \left(u^* + h_R\sqrt{\dfrac{g(h^* + h_R)}{2h^* h_R}}\right)t \\ W_R & si\ x > \left(u^* + h_R\sqrt{\dfrac{g(h^* + h_R)}{2h^* h_R}}\right)t \end{cases}$$

avec

$$W_1 = \begin{pmatrix} \dfrac{1}{9g}\left(u_L + 2\sqrt{gh_L} - \dfrac{x}{t}\right)^2 \\ \dfrac{1}{3}\left(u_L + 2\sqrt{gh_L} + 2\dfrac{x}{t}\right) \end{pmatrix}$$

et h^* et u^* solutions du système :

$$\begin{cases} u^* = u_L + 2\sqrt{g}\left(\sqrt{h_L} - \sqrt{h^*}\right) \\ u_R = u^* + (h_R - h^*)\sqrt{\dfrac{g(h^* + h_R)}{2h^* h_R}} \end{cases}$$