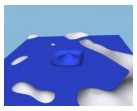
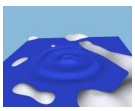
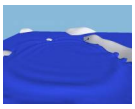




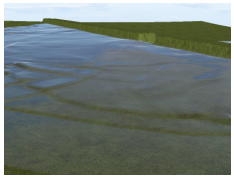
Animation physique d'une surface d'eau

- Equation de Saint-Venant ou Shallow Water Equation
- TP : OpenGL + OpenCL

Equation de Saint-Venant

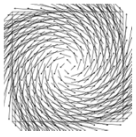
Equations de Saint-Venant ou Shallow Water equations

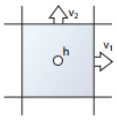
- Dérivée des eq. de Navier-Stokes
- Real Time Physics, Siggraph 2008 class.
- décrit les écoulements naturels : surface libre et eau peu profonde
- Grille 2D Approche Eulerienne (!= particule, approche Lagrangienne)

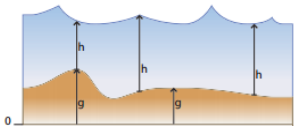



Equation de Saint-Venant

- Grille 2D
- h hauteur du fluide
- g hauteur du sol
- n hauteur du fluide sur le sol : n = h–g.
- v(v_1,v_2) vitesse du fluide dans le plan horizontal

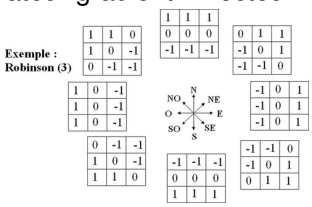

Champ de vecteur v





Préambule

- Opérateur gradient = vecteur 2D $\vec{\nabla} = \left(\frac{\partial}{\partial x} \quad \frac{\partial}{\partial y}\right)$



- Operateur Div = sommet des termes du gradient

$$\nabla \cdot \boldsymbol{A} \equiv \mathrm{div}\boldsymbol{A} = \frac{\partial A_x}{\partial x} + \frac{\partial A_y}{\partial y}$$

Equation de Saint-Venant : intuitif

- Equations de Saint-Venant (Shallow Water)

$$\frac{\partial \eta}{\partial t} + (\nabla \eta)\mathbf{v} = -\eta \nabla \cdot \mathbf{v} \qquad (1) \text{ avec } \nabla \cdot \boldsymbol{A} \equiv \mathrm{div}\boldsymbol{A} = \frac{\partial A_x}{\partial x} + \frac{\partial A_y}{\partial y}$$

$$\frac{\partial \mathbf{v}}{\partial t} + (\nabla \mathbf{v})\mathbf{v} = a_n \nabla h\,, \qquad (2)$$

- a_n : accélération verticale du fluide (gravité)=9.81
- (1) s'occupe de la variation de quantité d'eau
- (2) s'occupe de la variation de la vitesse

- Partie gauche : calculer par advection

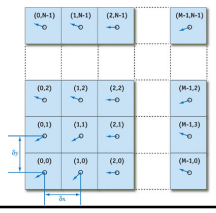
L'advection correspond au transport d'une quantité (scalaire ou vectorielle) par un champ vectoriel.

Equation de Saint-Venant : advection

- Equation de Saint-Venant (Shallow Water)

$$\partial \eta / \partial t + (\nabla \eta)\mathbf{v} = -\eta \nabla \cdot \mathbf{v} \qquad (1)$$
$$\partial v_1 / \partial t + (\nabla v_1)\mathbf{v} = a_n \nabla h \qquad (2)$$
$$\partial v_2 / \partial t + (\nabla v_2)\mathbf{v} = a_n \nabla h\,. \qquad (2')$$

- Partie gauche : calculer par advection



L'advection correspond au transport d'une quantité (scalaire ou vectorielle) par un champ vectoriel.

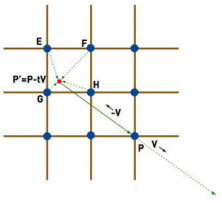
Equation de Saint-Venant : advection

- Equation de Saint-Venant (Shallow Water)

$$\partial\eta/\partial t + (\nabla\eta)\mathbf{v} = -\eta\nabla\cdot\mathbf{v} \quad (1)$$
$$\partial v_1/\partial t + (\nabla v_1)\mathbf{v} = a_n\nabla h \quad (2)$$
$$\partial v_2/\partial t + (\nabla v_2)\mathbf{v} = a_n\nabla h\,. \quad (2')$$

- Partie gauche : calculer par advection

Advect$(s,\mathbf{v})$
(1) **for** $j = 1$ **to** $n_2 - 1$
(2) **for** $i = 1$ **to** $n_1 - 1$
(3) $\mathbf{x} = (i\cdot\Delta x, j\cdot\Delta x)$
(4) $\mathbf{x}' = \mathbf{x} - \Delta t\cdot v(\mathbf{x})$
(5) $s'(i,j) = \text{interpolate}(s,\mathbf{x}')$
(6) **endfor**
(7) **endfor**
(8) **return**(s')



Equation de Saint-Venant : intuitif

- Equations de Saint-Venant (Shallow Water)

$$\frac{\partial\eta}{\partial t} + (\nabla\eta)\mathbf{v} = -\eta\nabla\cdot\mathbf{v} \quad (1)$$
$$\frac{\partial\mathbf{v}}{\partial t} + (\nabla\mathbf{v})\mathbf{v} = a_n\nabla h\,, \quad (2)$$

- (1) correspond à la variation de quantité d'eau
  - Variation de la quantité d'eau = dn/dt = -n∇.v = en fonction du champ de vitesse, on fait le bilan des arrivées des départs en eau
  - Si plus d'eau part, que d'eau arrive (∇.v>0), ca descend.
  - Et inversement.

Eq. Saint-Venant : variation quantité d'eau

- Equation de Saint-Venant (Shallow Water)

$$\partial\eta/\partial t + (\nabla\eta)\mathbf{v} = -\eta\nabla\cdot\mathbf{v} \quad (1)$$
$$\partial v_1/\partial t + (\nabla v_1)\mathbf{v} = a_n\nabla h \quad (2)$$
$$\partial v_2/\partial t + (\nabla v_2)\mathbf{v} = a_n\nabla h\,. \quad (2')$$

- Partie gauche : calculer par advection
- (1) donne variation de n donc n' = n – dt. (n∇.v)

Update-height$(\eta,\mathbf{v})$
(1) **for** $j = 1$ **to** $n_2 - 1$
(2) **for** $i = 1$ **to** $n_1 - 1$
(3) $\eta(i,j) -= \eta(i,j)\cdot\left(\frac{(v_1(i+1,j)-v_1(i,j))}{\Delta x} + \frac{(v_2(i,j+1)-v_2(i,j))}{\Delta x}\right)\Delta t$
(5) **endfor**
(6) **endfor**
(7) **return**(η')

Equation de Saint-Venant : intuitif

- Equations de Saint-Venant (Shallow Water)

$$\frac{\partial\eta}{\partial t} + (\nabla\eta)\mathbf{v} = -\eta\nabla\cdot\mathbf{v} \quad (1)$$
$$\frac{\partial\mathbf{v}}{\partial t} + (\nabla\mathbf{v})\mathbf{v} = a_n\nabla h\,, \quad (2)$$

- (2) correspond à : m.a = m.dv/dt = ∑F
  - Variation de vitesse
  - Ici F=a∇h = gravité dans la direction « vers moins d'eau »
  - le fluide subit une force dans la direction : de plus d'eau vers moins d'eau

Eq. de Saint-Venant : variation de v

- Equation de Saint-Venant (Shallow Water)

$$\partial\eta/\partial t + (\nabla\eta)\mathbf{v} = -\eta\nabla\cdot\mathbf{v} \quad (1)$$
$$\partial v_1/\partial t + (\nabla v_1)\mathbf{v} = a_n\nabla h \quad (2)$$
$$\partial v_2/\partial t + (\nabla v_2)\mathbf{v} = a_n\nabla h\,. \quad (2')$$

- (2)(2') donnent variation de v donc
  - v1 = v1 + $a_n\nabla h_1$ avec ∇h_1 = dh/dx
  - v2 = v2 + $a_n\nabla h_2$ avec ∇h_2 = dh/dy

Update-velocities(h,v_1,v_2,a)
(1) **for** $j = 1$ **to** $n_2 - 1$
(2) **for** $i = 2$ **to** $n_1 - 1$
(3) $v_1(i,j) += a(\frac{h(i-1,j)-h(i,j)}{\Delta x})\Delta t$
(5) **endfor**
(6) **endfor**
(7) **for** $j = 2$ **to** $n_2 - 1$
(8) **for** $i = 1$ **to** $n_1 - 1$
(9) $v_2(i,j) += a(\frac{h(i,j-1)-h(i,j)}{\Delta x})\Delta t$
(10) **endfor**
(11) **endfor**

Saint-Venant : intégration

- Shallow-water-step(h,v,g)
  - (1) n = Advect(n,v)
  - (2) v1 = Advect(v1,v)
  - (3) v2 = Advect(v2,v)
  - (4) n' = Update-height(n,v)
  - (5) h = n'+g
  - (6) Update-velocities(h,v1,v2)
  - (7) n=n'

➔ TP en C/C++

OpenCL (wikipedia)

- **OpenCL** (**Open** **C**omputing **L**anguage) est la combinaison d'une API et d'un langage de programmation dérivé du C, proposé comme un standard ouvert par le Khronos Group.
- OpenCL est conçu pour programmer des systèmes parallèles hétérogènes comprenant par exemple à la fois un CPU multi-cœur et un GPU.
- OpenCL propose donc un modèle de programmation se situant à l'intersection naissante entre le monde des CPU et des GPU, les premiers étant de plus en plus parallèles, les seconds étant de plus en plus programmables.

OpenCL

- OpenCL can accelerate code by a factor 10 or more
- OpenCL is an open standard
- OpenCL can help save power
- OpenCL can save you hardware cost
- OpenCL adoption is ramping up rapidly
- OpenCL may be used as the basis for generating custom hardware
- The OpenCL C99 language is based on C
- OpenCL can be used from a variety of host languages
- It is easy to start with OpenCL
- OpenCL is platform independent

http://www.amdahlsoftware.com/ten-reasons-why-we-love-opencl-and-why-you-might-too/

OpenGL / OpenCL : le TP

- Le code de départ
  - Carte de hauteur dans une texture 2D (GL)
  - Affichage GL avec un vertex shader
    - Vertex.glsl
  - kernel OpenCL modifie la texture
    - Texture 2D = vu comme une 'image2d'
    - CLWater.cl
    - Kernel = fonction appelée sur chaque case du tableau

OpenCL : un kernel

```
__kernel void shallowWaterInit( const int dimD,
                                __global __write_only image2d_t DD0 )
{
        const int x = get_global_id(0);
        const int y = get_global_id(1);

        if (x>=dimD) return; if (y>=dimD) return;
        const sampler_t sampler =
CLK_NORMALIZED_COORDS_FALSE | CLK_ADDRESS_CLAMP |
CLK_FILTER_NEAREST;

        float4 pixel={0,0,0,0};
        write_imagef( DD0, (int2)(x,y), pixel);
}
```


OpenCL : l'appel

```
clSetKernelArg( m_kernelShallowWater, 0, sizeof(Dim), (void*)&Dim);
clSetKernelArg( m_kernelShallowWater, 1, sizeof(D0), (void*)&D0);

const size_t localWorkSize[] = { LocalWorkSize, LocalWorkSize };
const size_t globalWorkSize[] = { Dim, Dim };

cl_uint workDim = 2;
clEnqueueNDRangeKernel ( m_queue, m_kernelShallowWaterInit,
                         workDim, NULL,
                         globalWorkSize, localWorkSize,
                         0, NULL, NULL);
```