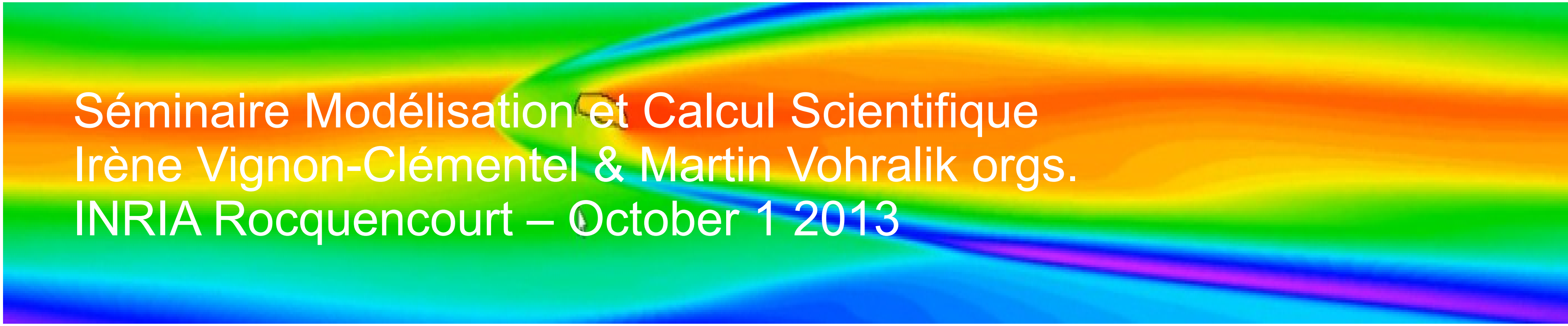


Séminaire Modélisation et Calcul Scientifique
Irène Vignon-Clémentel & Martin Vohralik orgs.
INRIA Rocquencourt – October 1 2013

Toward live CFD-computing interaction and visualization using GPU acceleration

Florian De Vuyst, Christophe Labourdette, Christian Rey

Centre de Mathématiques et de leurs Applications CMLA, CNRS UMR 8536

et

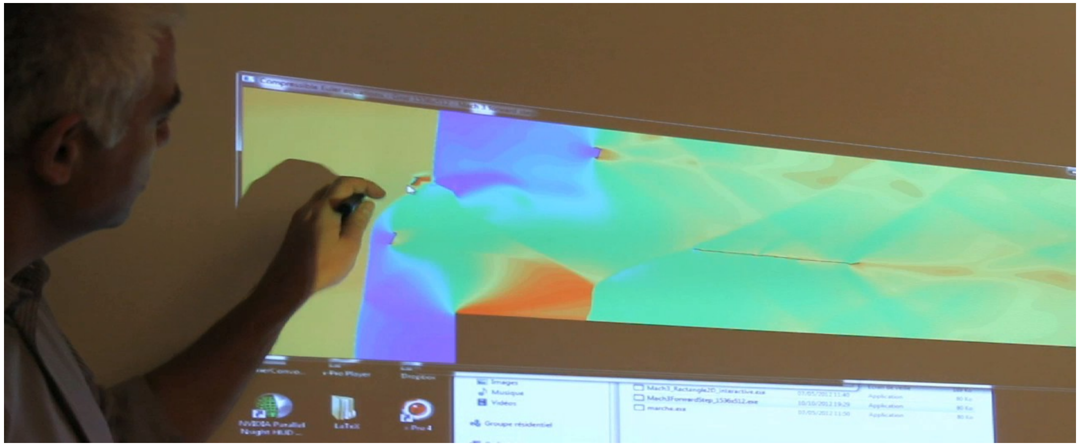
NVIDIA CUDA Research Center ENS CACHAN

devuyst@cmla.ens-cachan.fr

Objectives

- Get PDE approximate solutions instantaneously (« real time », co-simulation)

- Be able to directly act on the computations



Possible tracks ...

- Reduced-order modeling (POD, PGD, reduced basis method)
- High performance (parallel) computing
- Efficient algorithms, new paradigms (Lattice Boltzmann, ...)

- Multilevel modeling, surrogates

- Parallel computing on workstations (GPU, coprocessors)

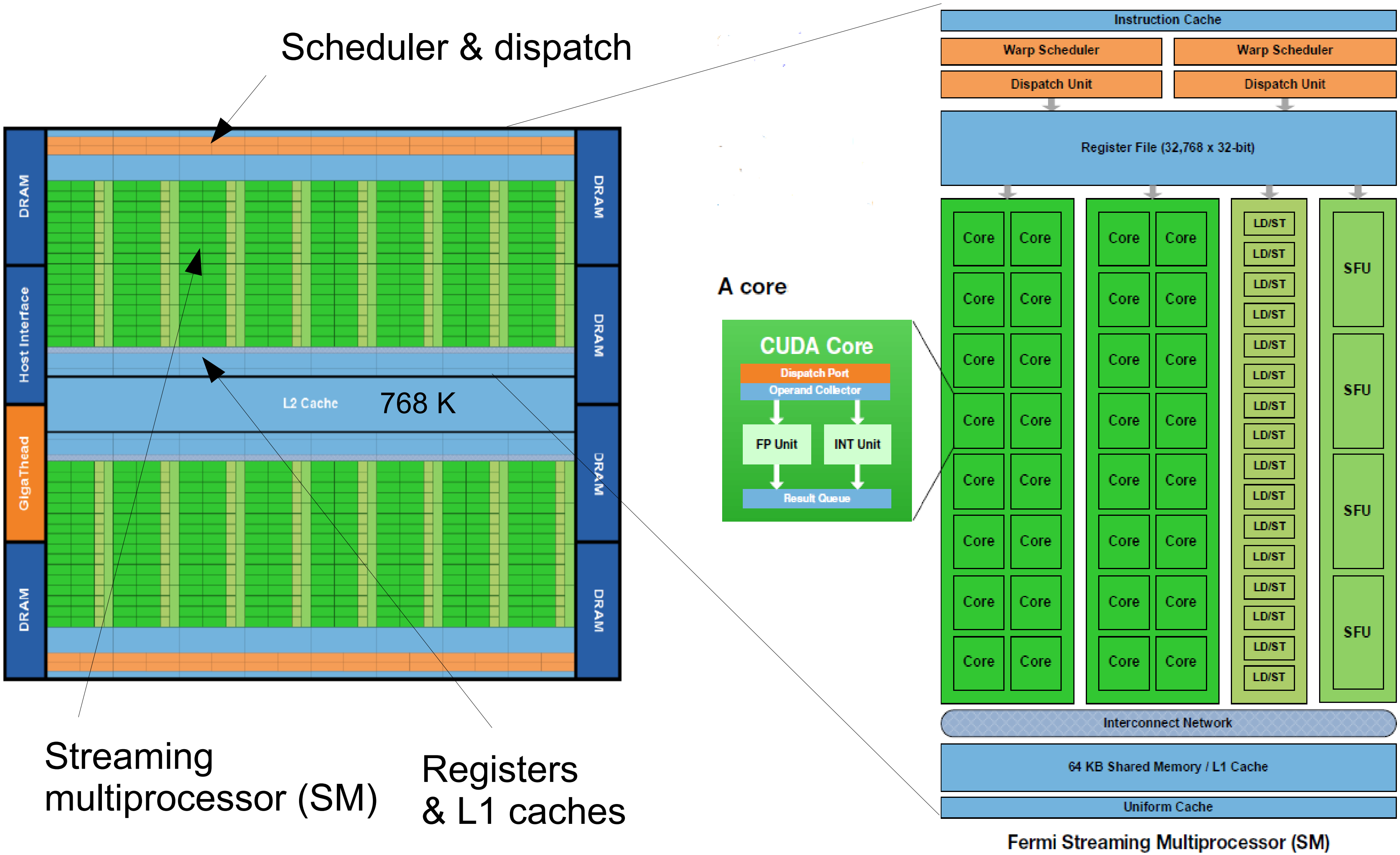
Toward GPU computing for PDE pbs – Outline

1. GPU hardware architecture
2. Performance drivers
3. Focus on Lattice Boltzmann Methods (LBM)
4. Real time flow interaction
5. Work in progress : LB thermal-fluid Boussinesq system
6. Suspension flows

Perspectives

1. GPU hardware architecture

nVIDIA GPU *FERMI* compute architecture



(2*16)*16 = 512 cores

(courtesy of *Nvidia*)

NVIDIA – KEPLER family (2013)

NVIDIA GeForce GTX 690 (game)

- 2 x 1536 cores, 300 W
- 8 streaming multiprocessors (SM)
- Memory bandwidth 192 GB/sec
- MEM 4 GB (2048 MB per GPU)
- DRAM bus memory 512-bit GDDR5
- 2x1.8 Tflops SP, 2x130 Gflops DP

NVIDIA TESLA K20X (HPC)

- **2688 cores**, 250W !
- 14 streaming multiprocessors (SM)
- Memory bandwidth 250 GB/sec
- MEM 6 GB
- 3.95 Tflops SP, **1.31 Tflops DP !**

2. GPU Performance drivers

Performance drivers

- Multiprocessor **occupancy**
- **Byte-per-flop** ratio (mem bandwidth vs FP operations)
- **Memory** management : registers, cache, coalesced read/write memory, fixed neighboring patterns reads/writes
- Warp divergence : be careful to trees of conditional branches
- Communication : host-to-device PCIe bus bottleneck

[Williams et al., Roofline : an insightful visual performance model for multicore architectures, Com. ACM, 2009]

Questions for the design of numerical methods :

- Explicit vs implicit ?
- Cartesian *vs* unstructured grid ?
- Order of accuracy *vs* grid size ?
- Specific *vs* undifferentiated treatment ? (interfaces, ...)
- Boundary conditions: embedded strategy ?
- How to achieve condensed stencil ?
- Operator splitting, alternating directions
- Change the model description ?
- Replace spatial derivatives by time derivative, particle derivative ? …

3. Focus on Lattice Boltzmann methods (LBM)

Lattice BGK (LBGK) model

Based on a simple discretization of the Boltzmann equation with BGK approximation for the collision term :

$$f(x + c\boldsymbol{e}_i\Delta t, \boldsymbol{e}_i, t + \Delta t) - f(x, \boldsymbol{e}_i, t) = \frac{1}{\tau}\left[f_i^{eq}(\rho, \boldsymbol{u}) - f(x, \boldsymbol{e}_i, t)\right]$$

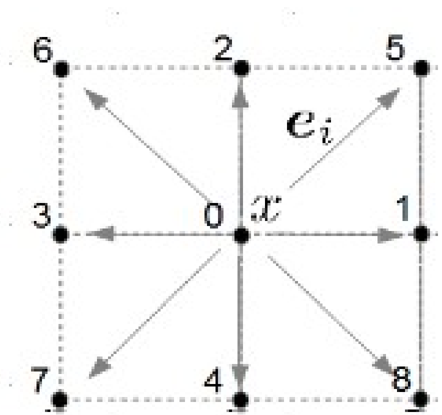
$$c = \frac{\Delta x}{\Delta t}.$$

Relaxation BGK « collision » term

Lattice Knudsen number

$$\boldsymbol{e}_0 = 0,\ \boldsymbol{e}_1 = (1,0),\ \boldsymbol{e}_2 = (1,1), ...$$

D2Q9 lattice



Moments:

$$\sum_{i\in\mathcal{S}} f(x, \boldsymbol{e}_i, t) = \rho(x, t),$$

$$\sum_{i\in\mathcal{S}} c\boldsymbol{e}_i\, f(x, \boldsymbol{e}_i, t) = \rho\boldsymbol{u}(x, t).$$

Lattice D2Q9 BGK model

- D2Q9 unknows $f_i(x,t) \equiv f(x, c\boldsymbol{e}_i, t),\ i \in \{0, ..., 8\}.$
- LB equation

$$f_i(x + c\boldsymbol{e}_i, t + \Delta t) = f_i(x,t) + \omega\ [f_i^{eq}(\rho(x,t), \boldsymbol{u}(x,t)) - f_i(x,t)]$$

$$\omega = \frac{1}{\tau}$$

- Requirements

$$\sum_{i \in \mathcal{S}} f_i^{eq}(\rho, \boldsymbol{u}) = \rho,$$

$$\sum_{i \in \mathcal{S}} c\boldsymbol{e}_i\ f_i^{eq}(\rho, \boldsymbol{u}) = \rho \boldsymbol{u},$$

$$\sum_{i \in \mathcal{S}} c^2 \boldsymbol{e}_i \otimes \boldsymbol{e}_i f_i^{eq}(\rho, \boldsymbol{u}) = \rho \boldsymbol{u} \otimes \boldsymbol{u} + pI$$

+ some discrete symmetry/ invariance conditions

Practical implementation « *stream*-and-*collide* »

1. *Streaming* step

$$\tilde{f}_i(x,t) = f_i(x + c\boldsymbol{e}_i, t)$$

2. *Collision* step

- Compute the moments $$(\rho, \rho\boldsymbol{u})(x,t) = \sum_{i\in\mathcal{S}} (1, c\boldsymbol{e}_i)\, \tilde{f}_i(x,t)$$
- Compute the equilibrium function $$\tilde{f}_i^{eq} = \rho\, w_i\, [...]$$
- Time advance

$$f_i(x, t+\Delta t) = \tilde{f}_i(x,t) + \omega \left[\tilde{f}_i^{eq}(\rho(x,t), \boldsymbol{u}(x,t)) - \tilde{f}_i(x,t) \right]$$

- **Explicit method**
- **Underlying cartesian grid**
- **Fixed (small) stencil**
- **Elementary operations**
- **Very easy to implement**

Particularly suitable for GPU computing

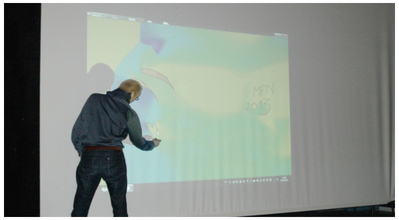
4. Real time visualization & flow interaction

Visualization & interaction

- Direct GPU compute/visualization binding (Pixel Buffer Object PBO) + OpenGL

- Dynamic mask array for adding/removing wall BC

- IR U-Pointer device for large screen interaction (seminars, conferences)



- Android tablets for controls & GUI. Use of OSC for bidirectional communication between GPU-workstation and tablet (coll. K. Labourdette).

5. Work in progress : LB thermal-fluid Boussinesq system

Thermal + CFD coupling : Boussinesq model

$$\nabla \cdot \boldsymbol{u} = 0,$$

$$\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla p - \frac{\mu}{\rho_0} \Delta \boldsymbol{u} = \boldsymbol{g} \left(1 - \beta(T - T_0)\right),$$

$$\partial_t T + \boldsymbol{u} \cdot \nabla T - \nabla \cdot (\kappa \nabla T) = 0,$$

$$+IC \;\; + BC.$$

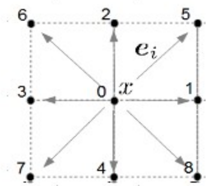
$$\mu, \alpha, \kappa > 0.$$

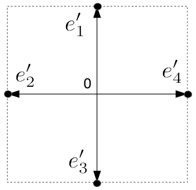
Convection, diffusion, source term, coupling.

2D Lattice Boltzmann discretization

$$c = \frac{\Delta x}{\Delta t}.$$

- D2Q9 LBGK lattice for fluid
- D2Q4 LBGK lattice for the thermal equation





Half-discretization (continuous in time) :

$$\partial_t f_i + c\boldsymbol{e}_i \cdot \nabla_x f_i = \frac{f_i^{eq} - f_i}{\tau \Delta t} - \frac{\boldsymbol{e}_i \cdot \boldsymbol{g}}{2c}\left(1 - \beta(T - T_0)\right)$$

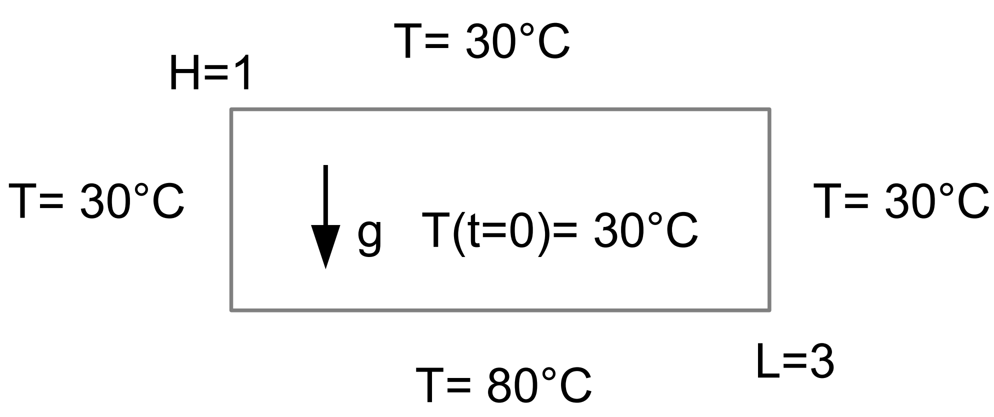
$$(\rho, \boldsymbol{u}) = \sum_{i=0}^{8} (1, \boldsymbol{e}_i)\, f_i, \;\; f_i^{eq} = f_i^{eq}(\rho, \boldsymbol{u})$$

$$\partial_t k_i + c\boldsymbol{e}'_i \cdot \nabla_x k_i = \frac{k_i^{eq} - k_i}{\tau' \Delta t}$$

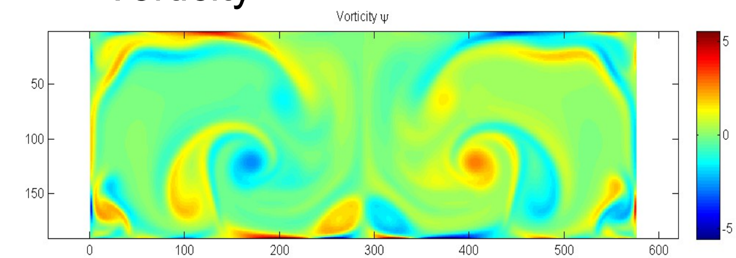
$$T = \sum_{i=1}^{4} k_i, \;\; k_i^{eq} = \frac{T}{4}\left(1 + 2\frac{\boldsymbol{u} \cdot \boldsymbol{e}'_i}{c}\right).$$

Validation of the method

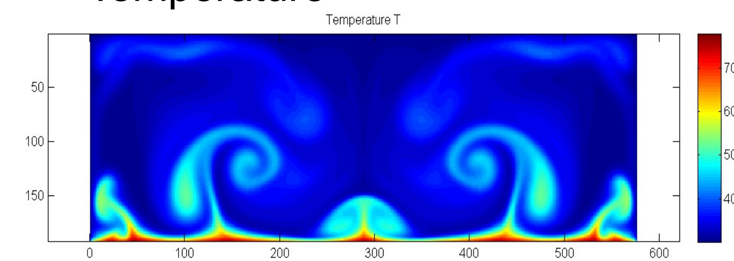
$$\boldsymbol{g} = 9.81,\ Re = 2300,\ Pr = 0.71,\ Ra = 3\,10^5,\ \nu = Re^{-1},\ \kappa = \frac{\nu}{Pr},\ \beta = \frac{Ra\,\nu\,\kappa}{gH^3\Delta T}$$



Vorticity

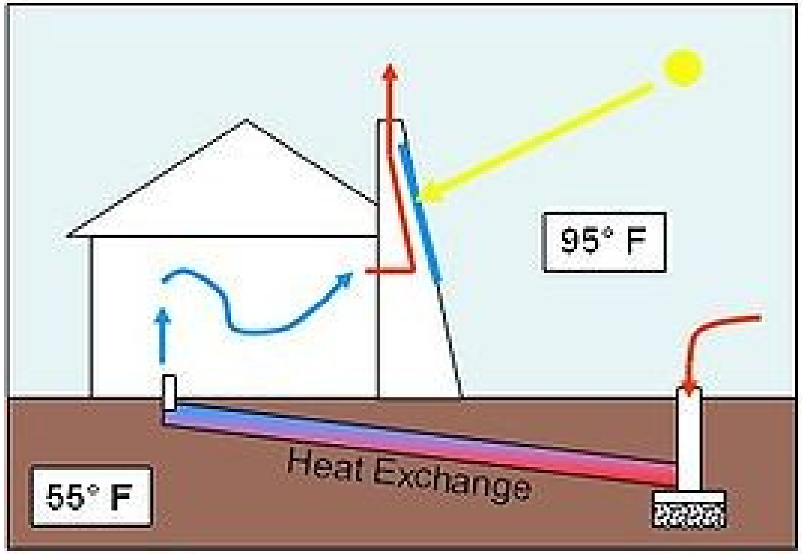


Temperature



Collaborators : S. Faure, L. Gouarin, B. Graille
(U. Paris-Sud Orsay)

Forseseen application : solar powered air conditioning

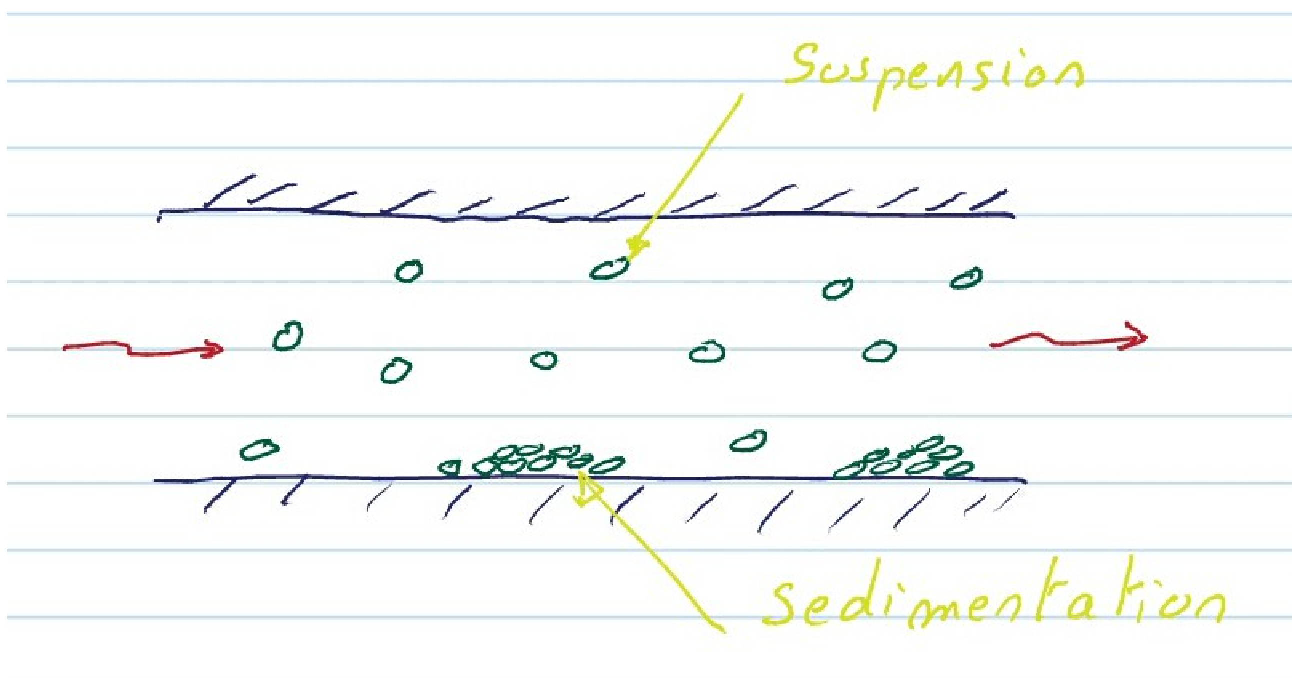


Design the chimney shape in order
to reduce recirculating zones at the top
and maximize the flow rate

(see [Chenier et al.], [Le Quéré, Sergent & co-workers] on the subject)

Work in progress

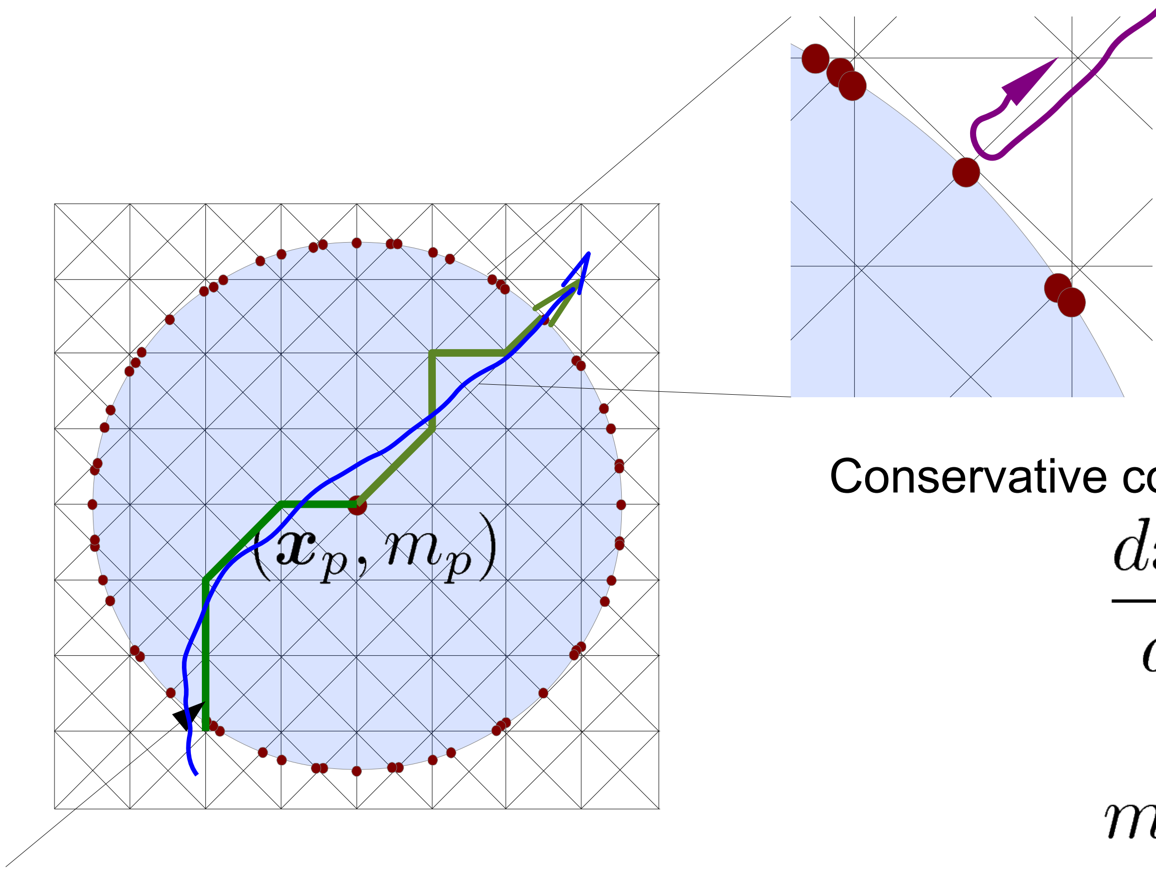
6. Suspension / sediment gravity flows



Suspension flows

Boundary conditions, flux formulation

[Dubois, Lallemand 2008]



Conservative coupling

$$\frac{d\boldsymbol{x}_p}{dt} = \boldsymbol{u}_p,$$

$$m_p \frac{d\boldsymbol{u}_p}{dt} = \int_{\partial S_p} \underline{\underline{\Sigma}}\,\underline{\nu}\, d\sigma.$$

Exact trajectory approximated by a broken-line lattice path (random choice stochastic method)

==> Geometric elements and intersecting nodes computed once-for-all

Tensor computed from 2nd-order discrete moments

Perspectives

- Euler-Euler multiphase flows (suspension)
- Immiscible fluids with moving interfaces
- NVIDIA outlook : 20 Tflops DP/GPU by 2018
  → real-time interactive 3D ?
- CPU-GPU convergence (be aware) ...

Acknowledgments

- NVIDIA CUDA Research Center ENS Cachan
- Institut fédératif FARMAN ENS Cachan
- Labex LMH Fondation Mathématique Hadamard FMJH