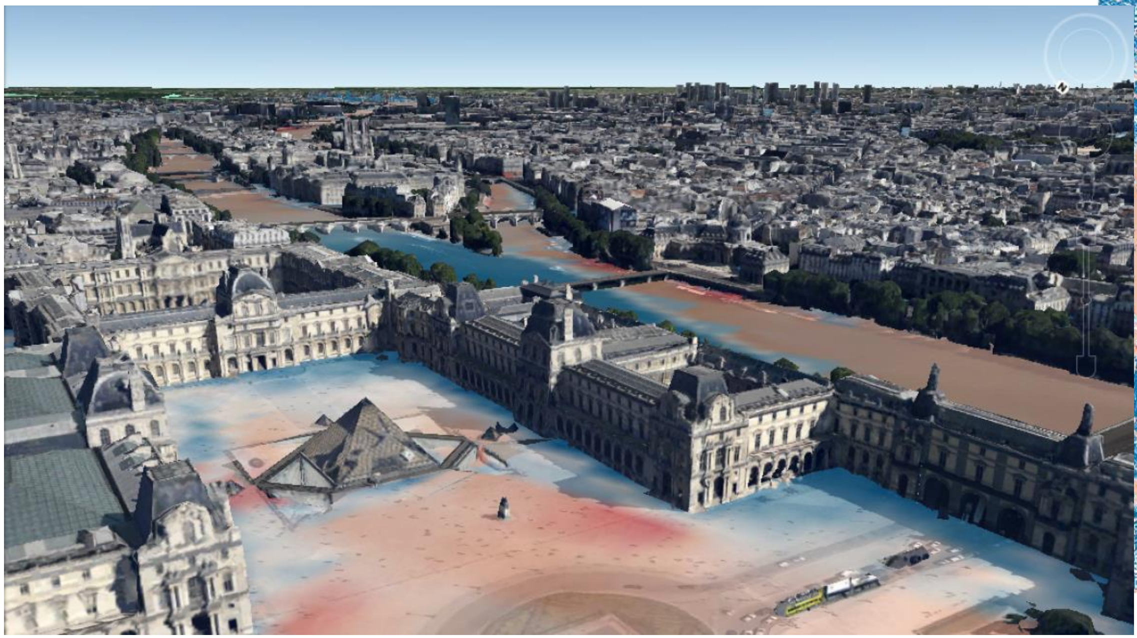
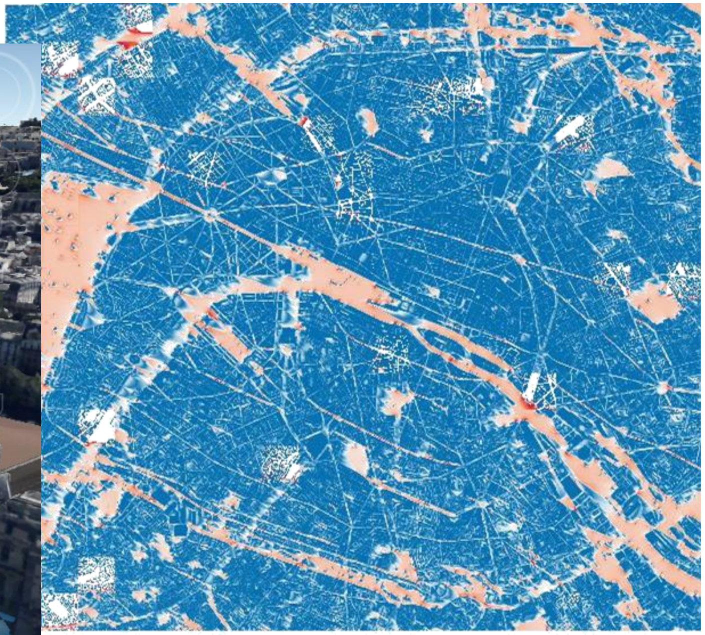
Introduction of momentum equations in Micro-SWIFT

Harmo'16 Conference, Varna

September 8-11 2013

Olivier OLDRINI[1], Maxime NIBART, Patrick ARMAND, Christophe OLRY, Jacques MOUSSAFIR and Armand ALBERGEL





[1]*E-mail: info@mokili.fr – Tel: +33 - 1 45 39 26 98*

Content

1. | Introduction

2. | Implementation of a fast solver

3. | Simulation results

4. | Conclusion

Introduction of momentum equations in Micro-SWIFT

1. Introduction

SWIFT / Micro-SWIFT are routinely used in nested / parallel mode

SWIFT / Micro SWIFT is a mass consistent interpolator over complex terrain. Micro SWIFT contains Rockle type modeling to take into account buildings.

SWIFT / Micro SWIFT capability can be used on a downscaling mode, called nested simulation.

Parallel version of MSS has also been developed to allow for operational handling of large build-up areas such as Paris .

The aim is to take into account more physic but keeping the low CPU cost

One particular application is parallel/nested simulations to handle malevolent / terrorist activities that may result in the atmospheric dispersion of noxious gases or particles in urban environment. Precise pressure diagnostic on very complex shape buildings may be needed for infiltration calculations.

To get more precise flow description around complex buildings, momentum constraint has been added to SWIFT / Micro-SWIFT.

The aim is to:

- Introduce more physic,
- Without increasing too much the computational cost

Introduction of momentum equations in Micro-SWIFT

2. Implementation of a fast solver

Traditional incompressible RANS equations are used to derive a stationary state

Equations are:

- Momentum:

$$\partial t\, U_i = -\, \partial_j (U_i U_j) - 1/\rho\, \partial_i P + \partial_j [\, \nu \, (\partial_i U_j + \partial_j U_i) + R_{ij})\,]$$

- Mass consistency:

$$\partial_i U_i = 0$$

With:

- U the wind
- P the pressure
- ρ the density, integrated in pressure from now on
- ν the kinematic viscosity
- R the Reynolds stress tensor

Turbulence closure is performed using simple mixing length

A mixing length is defined as:

$$l_{mix} = \kappa\, d_b$$

The turbulent viscosity ν_t is derived through:

$$\nu_t = l_{mix}^2 \sqrt{(S_{ij} S_{ij})}$$

With:

- S the deformation tensor: $S_{ij} = 1/2\, (\partial_i U_j + \partial_j U_i)$
- κ the von Karman constant
- d_b the distance to solid boundaries

Hence the momentum equation becomes:

$$\partial_t U_i = -\, \partial_j (U_i U_j) - \partial_i P + \partial_j [(\nu + \nu_t)(\partial_i U_j + \partial_j U_i)]$$

ARIA TECHNOLOGIES

Mokili Conseil en Environnement

Incompressibility uses artificial compressibility approach

Mass consistency:

$$\partial_i U_i = 0$$

Is substituted by a time varying equation for pressure:

$$1 / \beta \, \partial_{\square} P = - \partial_i U_i$$

Since we are solving for steady state, the artificial compressibility reduces to traditional incompressibility once convergence is reached.

A regular horizontal mesh with terrain following coordinates is used

Terrain following coordinates:

$X = x$

$Y = y$

$s = (H - z) / (H - z_g)$ with H the domain top and z_g the ground

Time discretization: second order explicit Adams-Bashforth scheme

Space discretization:

- Advection part: upwind scheme
- Diffusion part: second order centered

Introduction of momentum equations in Micro-SWIFT

3. Simulation results

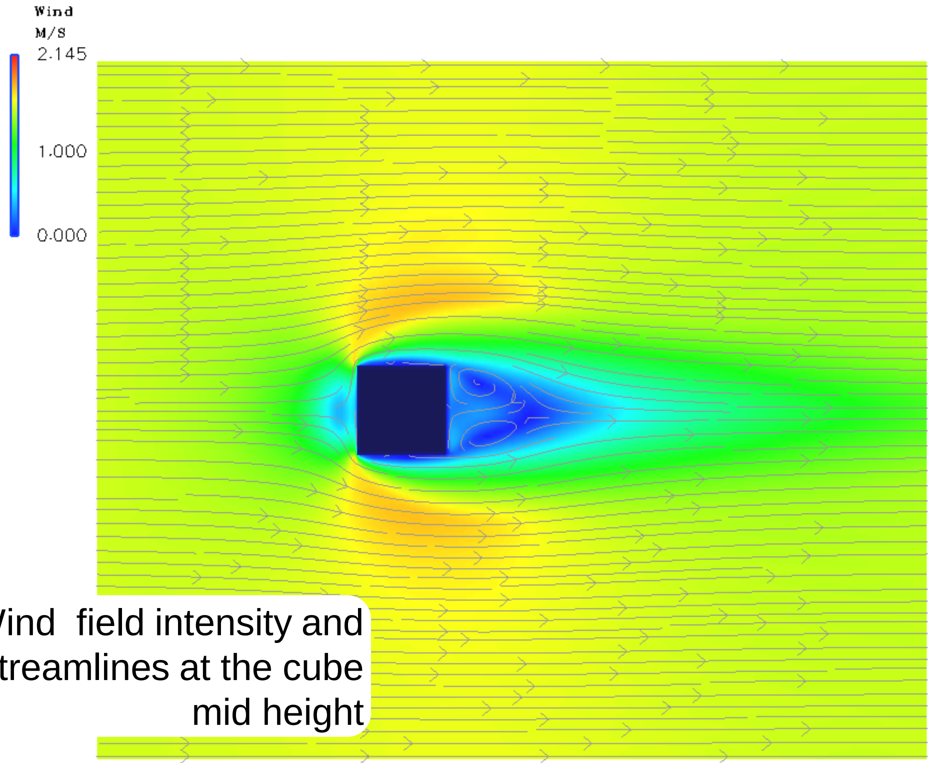
0.7 million nodes mesh on an isolated cube solved in 16mn

Configuration:

- 20m cube
- Domain: 191 x 161 x 23 (~ 0.7M nodes), metric resolution in horizontal, domain top at 60m
- Wind: academic log profile with 1.5m/s at 10m
- Micro-SWIFT wind field used as initial guess

With 2000 iterations of wind:

- Residual: 5.E-5
- 16mn on single Intel Core i5, 2.6GHz laptop



Wind field intensity and streamlines at the cube mid height

Comparisons to wind tunnel experiments have also been performed

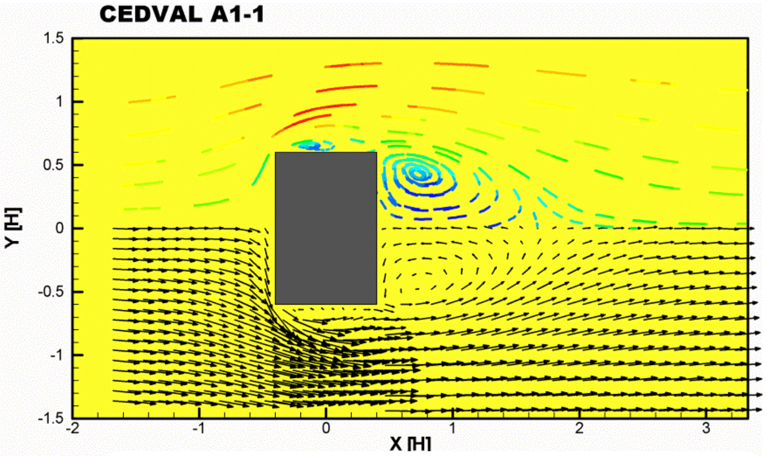
Configuration:

- Rectangular obstacle: 20m x 30m x 25m
- Inflow wind speed at building top height: 4m/s

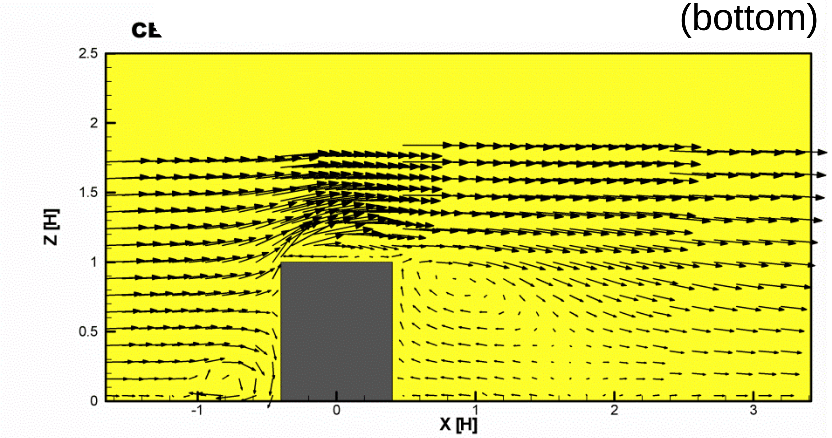
Data obtained for Hamburg University wind tunnel database

Mesh:

- 2m resolution
- 190 x 101 x 27 (~0.5 M) nodes



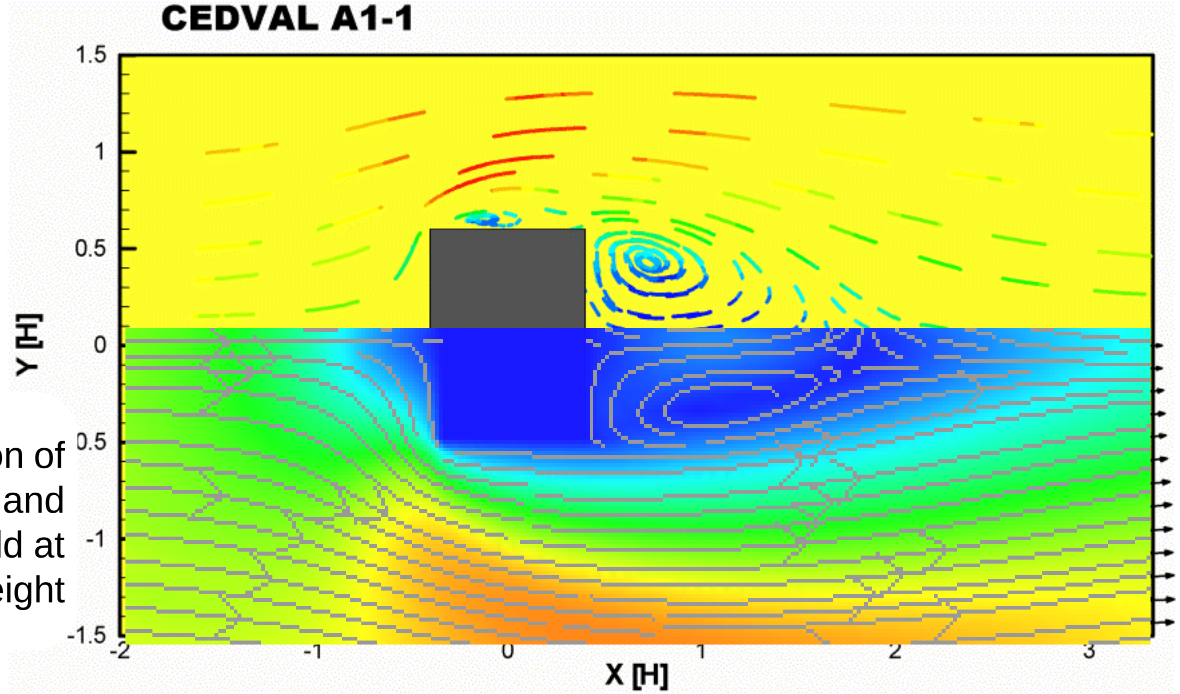
Experimental wind field at 7m height (top) and vertical median plane (bottom)



SWIFT-Momentum tends to over estimate vortex extension

Computation takes 14mn on Intel Xeon 5660 2,8GHz

Downwind vortex extension tends to be over estimated



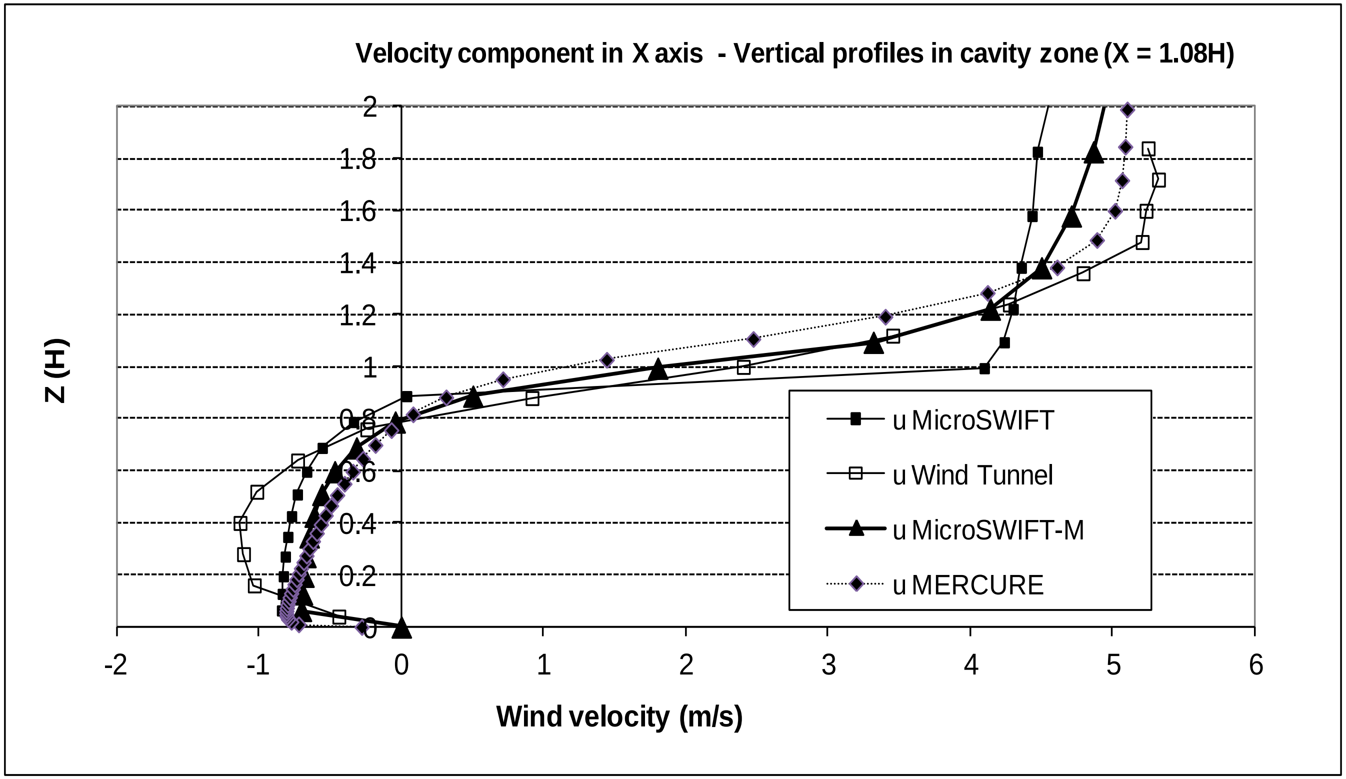
Comparison of experimental (top) and model (bottom) wind field at 7m height

SWIFT-M gives results closer to k-eps CFD

Comparison to Mercure k-eps CFD, Micro-SWIFT and experiment.

SWIFT-M gives an intermediate solution between Mercure and Micro-SWIFT

Wind profile at X = 1.08 height downwind of building for wind tunnel , Micro-SWIFT, Mercure and SWIFT-M

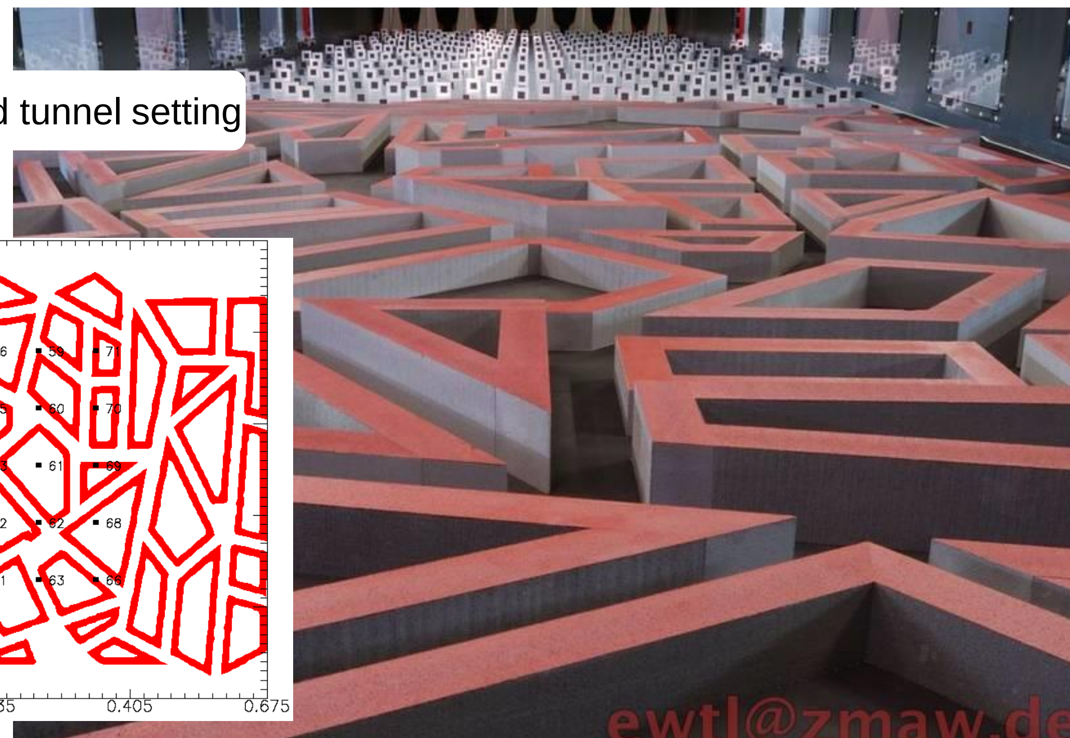


Preliminary testing has been performed on Michelstadt wind tunnel experiment both for CPU estimation and wind flow quality

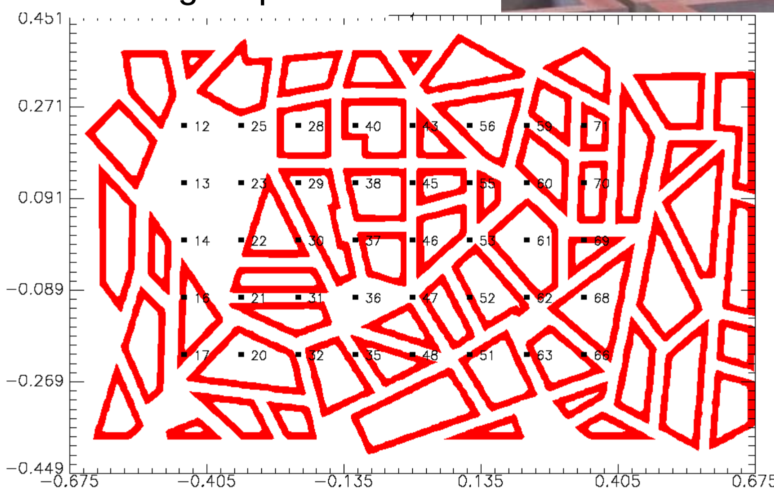
Michelstadt: wind tunnel academic city center defined in the COST framework

Mesh: 3m resolution in horizontal, 533 x 309 x 26 (~ 4.2 M) nodes

Michelstadt wind tunnel setting



Buildings top view



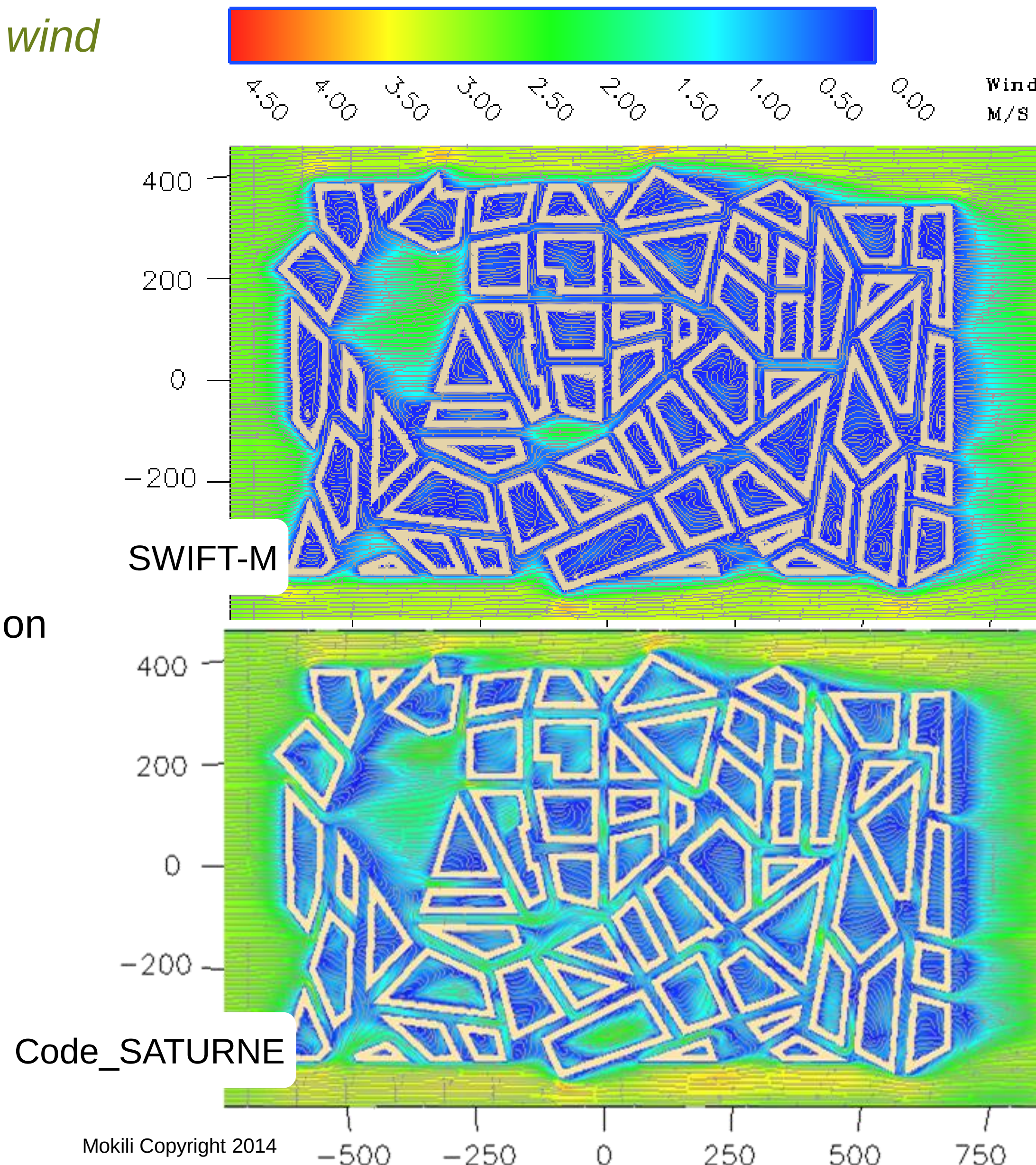
More flow friction leads to slower wind speed in streets

Preliminary results

CPU cost:

- SWIFT-M: 58 mn on a single Intel Xeon 5660 2.8 GHz processor
- Code_SATURNE: ~ 3h on 24 Intel Xeon 5680 3.3 GHz processors

Horizontal slice at 6m

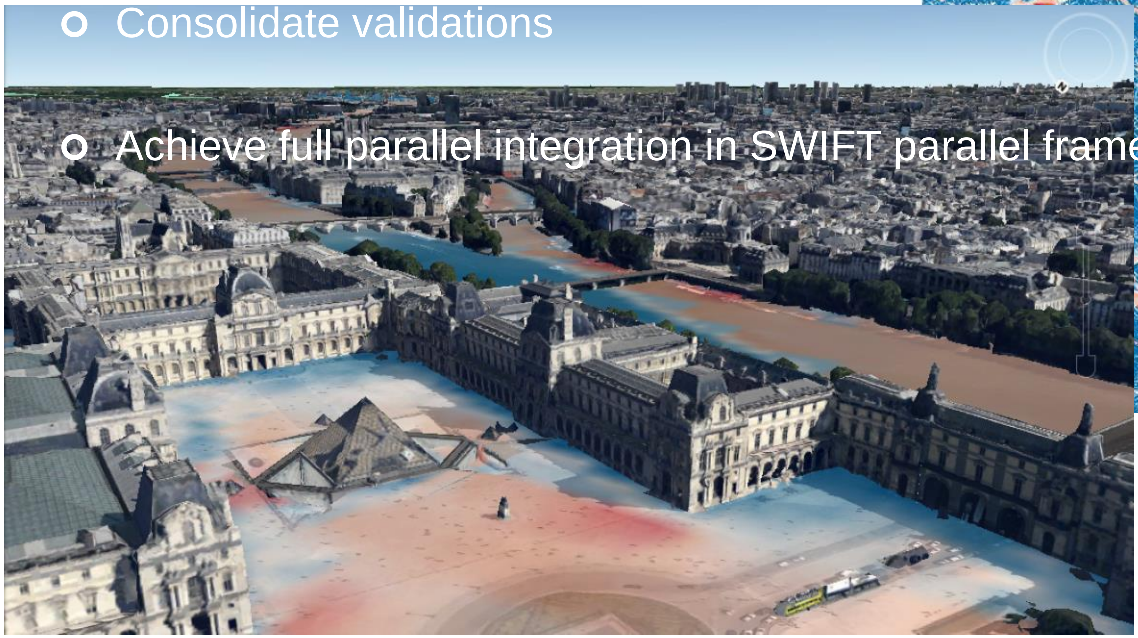


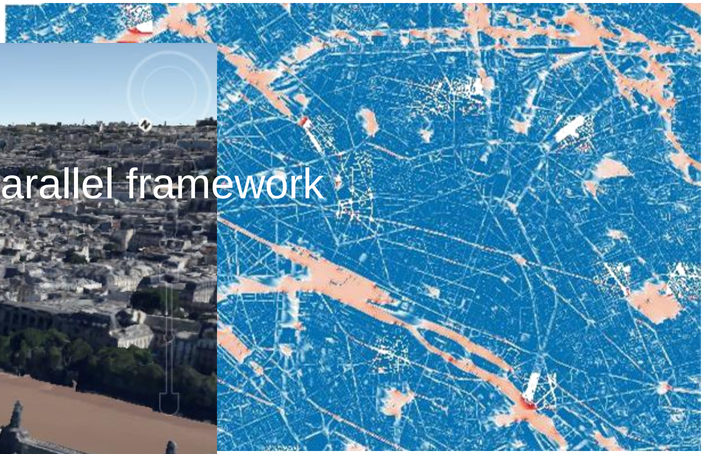
Introduction of momentum equations in Micro-SWIFT

4. Conclusions

SWIFT-M has low CPU constraints with more physic

- SWIFT-M allows us to take into account momentum constraint with cheap CPU overhead.
- Preliminary testing shows satisfactory agreement with full k-eps CFD on simple test cases.
- Next steps:
  - Consolidate validations
  - Achieve full parallel integration in SWIFT parallel framework





50mn SWIFT-M computation over whole Paris area with domain tiling

Thank you for your attention

SWIFT-M computation over whole Paris area with domain tiling: Details of wind field near Tour Eiffel

Wind speed (m/s)

- 0.000000
- 0.750000
- 1.500000
- 2.250000
- 3.000000

www.mokili.fr

MOKILI

Conseil
en Environnement