H14-122

APPLICATION OF PMSS, THE PARALLEL VERSION OF MSS, TO THE MICRO-METEOROLOGICAL FLOW FIELD AND DELETERIOUS DISPERSION INSIDE AN EXTENDED SIMULATION DOMAIN COVERING THE WHOLE PARIS AREA

Christophe Duchenne[1], Patrick Armand[1], Olivier Oldrini[2], Christophe Olry[3] and Jacques Moussafir[3]

[1]CEA, DAM, DIF, F-91297 Arpajon, France
[2]MOKILI, F-75014 Paris, France
[3]ARIA Technologies, F-92100 Boulogne-Billancourt, France

Abstract: The development of PMSS, the parallel version of the Micro-SWIFT-SPRAY suite, allows now to process 3D flow calculation, atmospheric dispersion of hazardous species and health impact assessment at the microscale, on extended urban areas, in a reduced computing time. Test cases on Paris show the feasibility of such calculations and the ability to obtain a very detailed representation of output fields, even far away from the release source. Evaluations of speedup prove that the parallelization of PSWIFT is very efficient both for tile's parallelization and timeframe's parallelization. Performances obtained with PSPRAY are quite good too but, as communications between cores are more important, parallelization may be less efficient for huge numbers of core.

Key words: *High performances computing, urban environment, micro-scale, Parallel-Micro-SWIFT-SPRAY, 3D Lagrangian dispersion, speedup, modelling and decision-support system.*

INTRODUCTION

Micro-SWIFT-SPRAY (MSS) modelling system has been designed for meteorological forecast and atmospheric transport modelling at the local scale, taking account of buildings. Micro-SWIFT is a diagnostic meteorological model and takes account of obstacles. It produces a mass consistent wind field using data from outputs of larger scales models or from a dispersed meteorological network. Temperature and humidity can also be interpolated. Micro-SPRAY is a Lagrangian dispersion model, based on a 'Monte Carlo' model. The motion of the particles is therefore obtained so as to reproduce the statistic features of turbulent flow. Special features of Micro-SPRAY are to take into account turbulence due to obstacles *via* Lagrangian time scales, to treat bouncing against obstacles, and to compute deposition on floor, walls and roofs.

Although MSS is able to simulate potentially toxic gases or particles dispersion inside dense urban environments in a reduced computing time compared with those of Computational Fluid Dynamics (CFD) softwares, a parallel version of MSS has been developed by ARIA Technologies, MOKILI and CEA (see Oldrini *et al.*, 2011). The two computational tools, Micro-SWIFT for the meteorological flow and Micro-SPRAY for the dispersion, have been independently parallelized to form the sequential suite PNSWIFT and PSPRAY. Parallelization is based on MPI programming system. The objective is both to reduce computing time and to allow to deal with huge computation grids, too large for the memory of a single core.

For PNSWIFT, two modes of parallelization coexists: the first one divides out computed times between available cores, the second one splits horizontal grid into tiles and attributes each tile to a core. If enough cores are available, a combination of both modes is possible. For PSPRAY, parallelization consists in managing the distribution of numerical particles between cores and managing active or inactive tiles. A master core is defined for each tile to compute concentrations and depositions at every synchronization time step.

Parallel Micro-SWIFT-SPRAY (PMSS) has been tested on various test cases, based on wind tunnel experiments or real scale releases. Oldrini *et al.*, 2011, show that differences between wind field performed by MSS and wind field performed by PMSS for the same configuration are not visually detectable. Moreover, differences on concentration fields are very slight and limited to tiles borders.

The aim of this work is to carry out PMSS on a very extended built area, covering the whole city of Paris. Considering a hypothetical test case, flow and concentration are performed inside each street and show the feasibility of this kind of calculation. In a second time, a speedup evaluation is done.

DESCRIPTION OF THE TEST CASE

The whole city of Paris is included inside a 12×10.5 km rectangle. To represent wind field with a good resolution inside each street of Paris, a 3-meter mesh size is chosen. It leads to a computation domain, for PNSWIFT, with 4001×3501 nodes for the horizontal grid. Vertically, 27 nodes are defined, with a logarithmic progression between ground and a height of 1000 meters. Finally, flow calculation is performed on a grid containing an amount of 380 millions nodes.

As a single core cannot handle with such a domain, it is tiled in 360 (20×18) sub-domains with $201\times201\times27$ nodes. A configuration with at least 361 cores is therefore required to solve this problem. For PSPRAY, the movement of numerical particles is performed inside the same computational grid but concentrations are projected on a domain using the same horizontal grid but a regular vertical grid with 20 nodes between ground and a height of 300 meters.

3D building data come from the BD TOPO® database, provided by the French National Geographical Institute (IGN) under a shapefile format, where all the buildings located inside the domain defined in Figure 1, are described with a little more than 50,000 polygons.

A pre-processor named SHAFT converts these polygons into a collection of about 600,000 triangular prisms written under an ASCII format, directly usable by PNSWIFT.

A real meteorological situation is simulated, based on forecasts provided by the operational meteorological forecast system MEDICIS (Achim *et al.*, 2010). Focused on France, MEDICIS generates, every six hours and among others, a forecast of the wind field at mesoscale with a one-hour resolution. For this work, a sequence of twelve time frames is selected, starting the 21st of September 2010 at 19h00, and is used to initiate and guide the flow calculation with PNSWIFT. So, twelve timeframes with a one-hour resolution are performed at the micro-scale.

Figure 1. Computation grid range (yellow contour).

For the simulation of atmospheric dispersion, two cases are considered. Both use 101 cores and same 3D wind fields. Case A considers a release of a gaseous chemical substance is considered from place de l'Etoile in Paris. Release is supposed to occur near the ground, during a 20-minute long period. Case B considers a release of a radioactive aerosol from Meynadier Street (near parc des Buttes-Chaumont in the North East of Paris). Release is supposed to occur near the ground, during a 2-minute long period. Concentration and deposition fields are recorded every minutes and simulation is limited to a two-hour long period (with the selected meteorological conditions, emitted clouds are completely outside from Paris in this time). An amount of 120 timeframes are therefore recorded.

TEST CASE RESULTS

In the configuration we have chosen, flow calculation with PNSWIFT produces 360 binary output files, each with twelve timeframes, which all take up about 260 Go memory space. Main orientation of the flow and its time evolution is shown on Figure 2. Flow near the ground, inside urban canopy, is much more complex and wind directions and wind velocities inside streets might be very different as mean wind, depending on the orientation of the streets and position and shape of buildings. PNSWIFT provides very detailed wind fields, which allow to represent recirculations and eddies occurring inside streets. Tiling has no effect on the computed meteorological flow, because influence of buildings on the wind field is considered by the algorithm, not only inside the tile where the buildings are located, but also outside, in adjacent tiles.

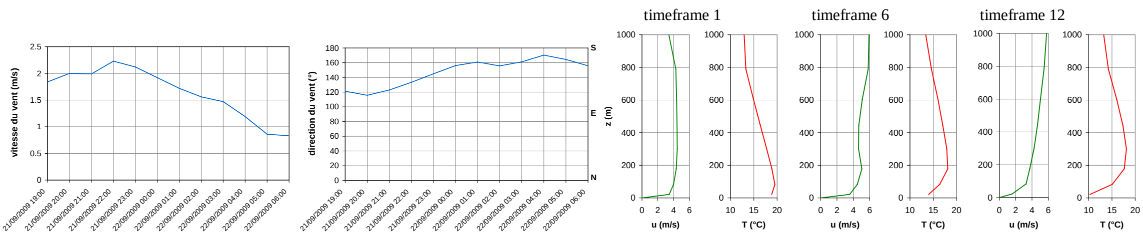

Figure 2. Time evolution of wind (velocity and direction) and vertical profiles near place de l'Etoile at different times.

Lagrangian model PSPRAY computes atmospheric dispersion using 3D flows performed by PNSWIFT. According to meteorological conditions, both releases, for case A and case B, lead to a cloud moving towards the North-West. In case A, 50 tiles see part of the cloud during the simulated period (Figure 3), while 83 tiles are impacted by part of the cloud in case B (Figure 4). Concentrations are recorded only for active tiles and only for the period where they are active. Binary results files are also compact for such a huge calculation (28 Go for case A and 46 Go for Case B).

PSPRAY performs very detailed concentration fields inside all streets of the computation domain. As concentration is computed by projection of the contribution of each numerical particle in the concentration grid, statistical effects occur in zones with few particles which affect the representation of the concentration field. Slight discontinuities appear at the borders of the cloud. As PSPRAY allows to deal with a huge amount of particles (21.6 millions for case A), relevant concentrations are obtained on a very extended area. Displacement of numerical particles from a tile to another has a slight effect on the concentration field at their interface, because particles arriving at the interface are frozen until the next synchronization time step, where they are transferred to the adjacent tile. As the frequency of synchronization is a user defined parameter, usually fixed at a high value, this effect is strongly reduced.

t_0+5min t_0+15min t_0+40min

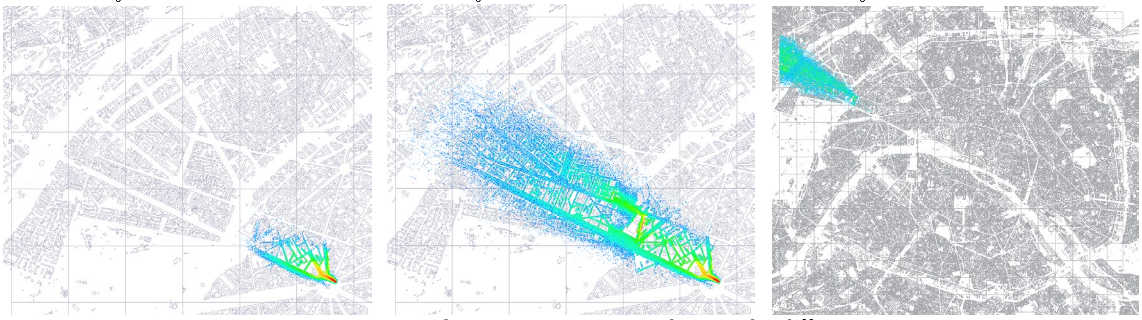

Figure 3. Case A: atmospheric concentrations near the ground at different times.

t_0+5min t_0+15min t_0+30min

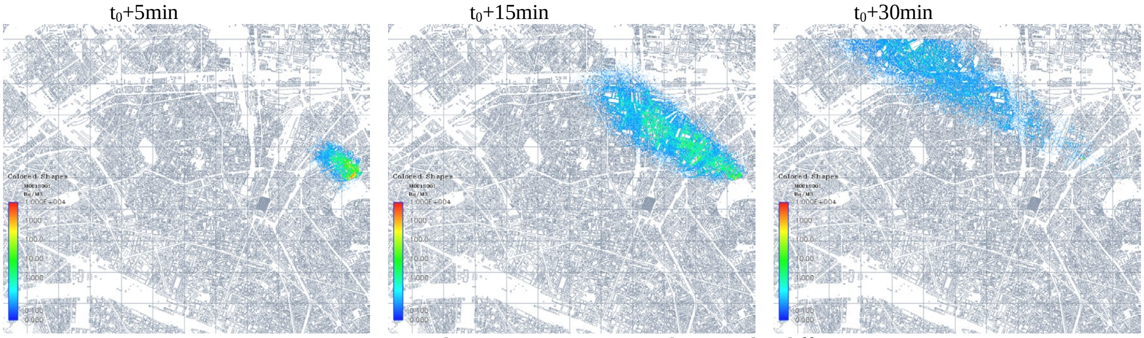

Figure 4. Case B: atmospheric concentrations near the ground at different times.

When dispersed specie is an aerosol, such as in case B, deposition can be performed on ground and on all accessible surfaces (façades and roofs). As for atmospheric concentration, deposition fields are very detailed.

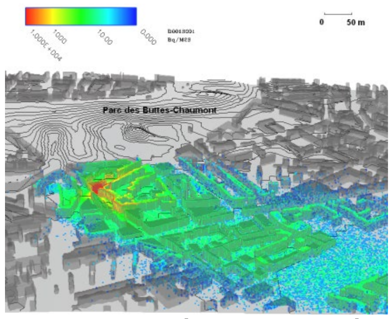

Figure 5. Case B: deposition on ground, façades and roofs once the cloud has gone.

Post-processing results of atmospheric concentrations or depositions allow to compute chemical dose or radiological impact depending on the nature of the emitted species. In case of a radioactive species, the module SPRAYSHINE (Armand *et al.*, 2011), which deals with multi-tiles domains in a parallel mode, can be activated to compute an accurate dose field due to radiations emitted by the cloud and by depositions on ground, façades and roofs. According to regulatory thresholds, danger zones can be defined and help authorities to decide the setting up of appropriate countermeasures.

PERFORMANCES OF PNSWIFT

Two ways exists to assess performances of a parallel code. Either the aim is, for a given problem, to reduce computing time and then, we have to compare computing time with one core (T_1) with computing time with several cores (T_n) to solve the same problem, or, the aim is to solve a bigger problem and then, we have to compare computing time with one core to solve a problem with a size X (T_1) with computing time with n cores to solve of problem with a size n×X (T_n). In both approaches, speedup (S_n) is defined by the same relation:

$$S_n = \frac{T_1}{T_n} \qquad (1)$$

As the objective of parallelizing Micro-SWIFT is both to reduce computing time and to allow to deal with huge computation grids, performances of PSWIFT are evaluated regarding these both approaches, the first one named Amdahl approach, and the second one named Gustafson approach.

Flow calculations of the case above are done for a various number of cores, from 10 to 701. When the number of cores increases, the domain covering the Paris area is cut in a larger number of tiles up, and tiles size decreases. Calculations are made using High Performances Computing resources of the Research and Technology Computing Center (CCRT – **http://www-hpc.cea.fr/en/complexe/ccrt.htm**). Cluster used is a 47.7 Tflops BULL Itanium cluster with 932 nodes (more than 7000 1.6 GHz Itanium cores). Computing time decreases from about one day, using 10 cores to compute twelve timeframes of 3D flow field covering the whole city of Paris, to less than half an hour using 701 cores (Figure 6).

The speedup evaluation to determine computing time in a standalone mode (T_1), that is to say using a single core. As the size of the problem is too big to do the calculation with a single core, T_1 is extrapolated from CPU time. Figure 6 show that CPU time increases with the number of cores (n), except for small values of n, where problems appear due to the big and inappropriate size of tiles. Extrapolation of the CPU time curve for n=1 gives a value of about 800,000 s. We choose even $T_1 = 750{,}000$ s to be sure not to overestimate speedup in the following evaluation.

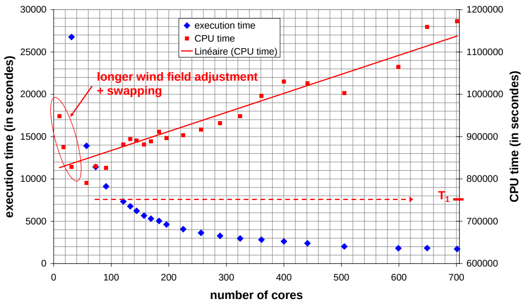

Figure 6. Computing and CPU time functions of number of cores.

The red curve in Figure 7a represent speedup performed from computing time of Figure 6. Speedup is worth 54 for n = 57, 102 for n = 121, 184 for n = 225 and 413 for n = 599. PSWIFT speedup is close to ideal speedup for n < 100. Speedup is still very good for some hundreds cores and continues to grow for n ≈ 700. Thus, parallelization by domain tiling is very efficient for PSWIFT.

Some user defined parameters have an influence on the performances of PSWIFT. Among them is the EPSILON parameter, which define the convergence criterion for the adjustment step of flow calculation. A more restrictive criterion increases the number of iterations needed to obtain a consistent wind field. It improves speedup as well, as shown in Figure 7a (green curve), but computing time is 30 to 50% longer.

Figure 7b compares speedups for parallelization in subdomains and parallelization in time frames and shows that there is almost no difference between the both ways of parallelization, in terms of performance.

Evaluating speedup with Gustafson approach implies to do calculation on a single reference tile with the standalone version of PSWIFT and to do calculation on n tiles, each having same size as the single size, with n cores. Figure 7c shows that speedup decreases from 0.7 to 0.3 when n increases from 30 to 700. Considering the reference case for Paris, with 360 tiles having 201×201 nodes, flow calculation with PSWIFT takes 2.5 more time than flow calculation on a single tile 201×201 nodes, with the standalone version. Such performances, compared to those obtained with the first approach, seem a little disappointing, but solving wind field on a tile inside a tiled domain is different as solving wind field on a single tile. In fact, we have to take account of buildings in adjacent tiles and it induces more calculations. Evaluated speedups are nevertheless pretty good.

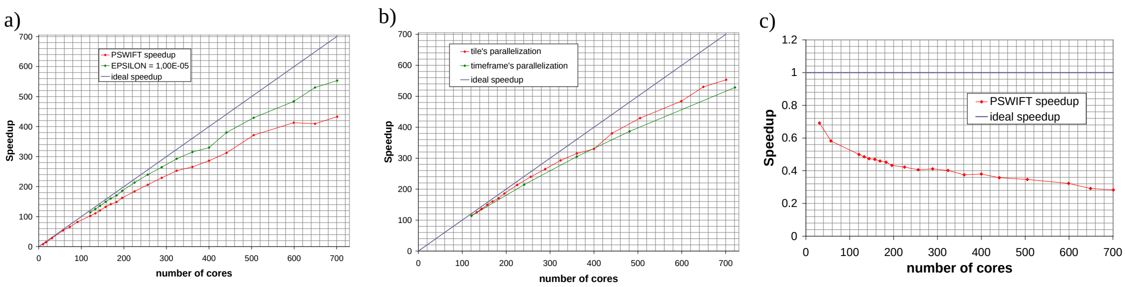

Figure 7. Speedup as a function of number of cores for different values of parameter EPSILON (a), for both modes of parallelization (b), and according to Gustafson's approach (c)

PERFORMANCES OF PSPRAY

Performances of PSPRAY, the parallel version of Micro-SPRAY, are assessed in relation to test case A. In this case, 50 tiles see a fraction of the cloud and at most 48 tiles at the same time. Dispersion calculation can be achieved only if at least 49 cores are carried out. Table 1 shows, for n going from 49 to 401, that computing times decrease a lot until n = 201 and stagnate for n > 201.

Table 1. Computing time

number of cores	computing time (s)
49	49,748
76	16,449
101	11,428
151	8,073
201	6,948
251	6,195
301	6,225
401	5,876

These computing times may seem a bit long but an amount of 21.6 millions numerical particles are simulated in case A, which is much more (about a decade) than the number of particles usually used in practice. As computing time for n = 1 cannot be assessed because such a calculation has no sense in this case, the speedup definition given by relation (1) cannot be applied. If we assume $T_1 = T_{49}$, speedup exceeds ideal speedup and it has no sense either. So another definition of the speedup has to be found, to represent in a realistic way performances of PSPRAY to reduce computing times.

Figure 8 shows the number of active tiles (N_{ta}) as a function of time during the simulation. For case A, $N_{ta\ max} = 48$. At $t = t_0+5$ min, $N_{ta} = 14$: at this time step, for a calculation with 49 cores, there are 14 cores allocated as master cores at these tiles and 34 cores to share out among tiles with a lot of number of particles to treat. At $t = t_0+30$ min, $N_{ta} = 47$: at this time step, for the same calculation with 49 cores, there are 47 cores allocated as master cores at these tiles and only 1 core to allocate to the tile with the most particles to treat. For the whole time of the simulation, the mean value of N_{ta} is equal to 37. With $n_{ta} = N_{ta}+1$ as the minimum number of cores needed to treat N_{ta} active tiles, we suggest to define speedup as:

$$S\left(\frac{n - n_{ta\ moy}}{n_{ta\ \max} - n_{ta\ moy}}\right) = \frac{T_{n_{ta\ \max}}}{T_n} \quad (2)$$

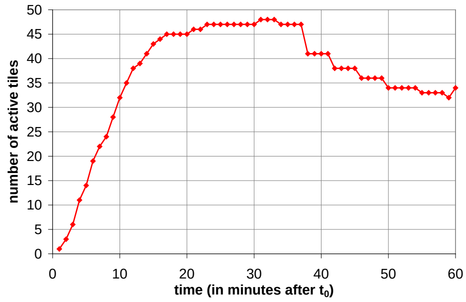

Figure 8. Number of active tiles as a function of time for test case A

Figure 9 shows speedup plotted according relation (2). PSPRAY speedup is close to ideal speedup for a ratio < 8, that is to say for $n_{cores} < 151$ in case A. At this step, each additional core allows to reduce significantly the maximum number of particles to treat per core. PSPRAY speedup does although not increase when ratio is over 15, that is to say for $n_{cores} > 251$. At this step, load-balancing is optimized and additional cores bring a small contribution in reducing computing time. Communications between cores increase then and become very significant for high numbers of cores.

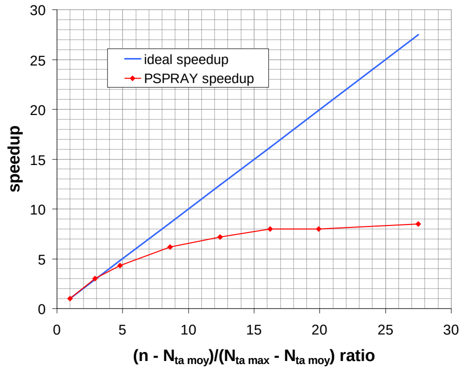

Figure 9. Speedup for PSPRAY

Finally note that evaluating PSPRAY performances in general is difficult because it depends, among others, on release (kinetic, number of sources, position in the grid), load-balancing frequency, and number of simulated particles.

CONCLUSION AND PERSPECTIVES

Modelling system PMSS is a sequential suite where a flow calculation with PSWIFT is followed by atmospheric dispersion using PSPRAY. Parallel mode allows significant reductions of computing time and calculations on huge urban areas, such as Paris, New York City or Tokyo.

Test cases on Paris have proven that PMSS is able to represent in a fine and very detailed way, atmospheric dispersion of hazardous radiological or chemical specie inside all the streets of this huge computation domain. Moreover, in the limit of tiles, wind field, atmospheric concentration and deposition on all accessible surfaces are not visibly influenced by the interface. Results are practically the same as if there was no interface at this place.

Performances of PSWIFT are very impressive, with high reductions of computing time even when many hundreds of cores are carried out. Evaluating PSPRAY speedup is not so simple as it depends on release's scenario and some user defined parameters, but results we obtain for the studied test case are pretty good.

Finally, building an operational meteorological forecast system at the microscale for big cities survey is henceforth possible. Computing times are compatible with such a system, which will provide automatic and continuously forecasts of the flow field. Atmospheric dispersion may also be processed on demand.

REFERENCES

Achim, P., H. Dupont, A. Leroy and P. Armand, 2010: An operational meteorological forecast system at mesoscale for radiological and chemical impact assessment, 13th International Conference on Harmonisation within Atmospheric Dispersion Modelling fir Regulatory Purposes, Harmo'13, Paris (France), June 1-4, 2010.

Armand, P., C. Olry, O. Oldrini, J. Carrère, C. Duchenne and A. Albergel, 2011: Simulation with Micro-SPRAY and post-processor SPRAYSHINE of the irradiationdue to a radioactive plume and its deposition on all accessible surfaces in the urban environment, 14th International Conference on Harmonisation within Atmospheric Dispersion Modelling fir Regulatory Purposes, Harmo'14, Kos (Greece), Oct. 2-6, 2011.

Oldrini, O., C. Olry, J. Moussafir, P. Armand and C. Duchenne, 2011: Development of PMSS, the parallel version of Micro-SWIFT-SPRAY, 14th International Conference on Harmonisation within Atmospheric Dispersion Modelling fir Regulatory Purposes, Harmo'14, Kos (Greece), Oct. 2-6, 2011.