H14-176

DEVELOPMENT OF PMSS, THE PARALLEL VERSION OF MICRO-SWIFT-SPRAY

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

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

Abstract: Micro-SWIFT-SPRAY is a fast transport and dispersion modelling system. It is designed for local scale and takes into account buildings. A parallel version of MSS has been developed by ARIA Technologies, MOKILI and CEA. MSS consists of SWIFT, a mass consistent nested wind field model, and SPRAY, a Lagrangian particle dispersion model. PMSS has been designed to decrease the computational time on very large, high resolution, domains with high number of particles. Parallelization has been successfully implemented both for splitting geographically big memory domains being split, and based on code specific properties to gain speed up. Geographic parallelization is achieved through classical Eulerian tile splitting. Most of the speed up is achieved on SWIFT based on the diagnostic property of the code: different time frames can be computed at the same time without any communication. Geographic parallelization is driven by master core: domain is being divided into sub domains. Specific geographic information, such as topography or roughness, is sent to relevant cores. Building data needed specific attention due to Röckle type algorithm to handle wind initialization. Communication and calculation between cores on adjacent tiles are performed in such way to allow results to be identical to the simulation without geographic parallelization. Transition to parallel SPRAY is handled smoothly: SPRAY uses domain decomposition inherited from SWIFT parallel computation. SPRAY speedup is achieved by particle splitting between cores. SPRAY allows loading in memory sub domains according to source locations at first and then containing active particles. SPRAY has very elaborated load balancing to provide sub domains containing numerous particles with maximum core power and is able to transition particles between sub domains. Quality of results is very good and performances, even with high number of cores, are in line with expectations as shown on several traditional MSS test cases. PMSS is designed to take advantage of computer power on a diversity of architectures, from multi core laptops to very large super computer cluster. Applications of PMSS are ranging from air quality modelling to emergency response purposes.

Key words: *Micro SWIFT SPRAY, mass consistency terrain following Röckle type model, Lagrangian dispersion modelling, parallel computing.*

INTRODUCTION

MSS, Micro-SWIFT-SPRAY (Moussafir et al., 2004 and 2007), is a fast transport and dispersion modelling system. It is designed for local scale and takes into account buildings. MSS consists of SWIFT and SPRAY used in urban mode (Micro SWIFT, Micro Spray).

Micro-SWIFT (Moussafir et al., 2004; Tinarelli et al., 2007) is an analytically modified mass consistent interpolator over complex terrain. Given topography, meteorological data and buildings, a mass consistent 3-D wind field is generated. It is also able to derive diagnostic turbulence parameters (namely the Turbulent Kinetic Energy, TKE, and its dissipation rate) to be used by Micro-SPRAY inside the flow zones modified by obstacles. Micro-SPRAY is an LPD (Lagrangian Particle Dispersion) model able to take into account the presence of obstacles. It directly derives from the SPRAY code (Anfossi et al., 1998; Carvalho et. al, 2002; Ferrero et al., 2001; Ferrero and Anfossi, 1998; Kerr et al., 2001; Trini Castelli et al., 2003; Tinarelli et al., 1994 and 2000). It is based on a 3-D form of the Langevin equation for the random velocity (Thomson, 1987).

MSS is used for domains with fine resolution or computations on a very long period of time. These usages tend to become more and more computationally intensive as the domains grow larger and larger, or the simulation period longer. Parallel MSS was designed to solve the problems related to intensification of computation.

SWIFT

Non parallel structure

SWIFT, the analytically modified mass consistent interpolator over complex terrain, performs nested calculations for wind, temperature and turbulence. Each nested domain is treated sequentially in a downscaling order. Data computed are transmitted between nests from larger scale domains to finer resolution domains.

A simulation at a certain nest level consists of the treatment of multiple independent time frames. For each time frame, SWIFT generates first guess wind and temperature 3D fields, starting from available measurements (sonic anemometer, sodar, lidar, …), larger scale model outputs (WRF, …) or larger scale SWIFT calculation. If obstacles such as buildings are available, wind flow is modified using analytically modified flow zones. After this interpolation step, the wind field is modified using a Poisson solver to get a mass consistent wind field that takes into account impermeability of topography and buildings. This step is the adjustment step. Diagnostic turbulence parameters are then derived from wind and temperature fields.

Parallel SWIFT

MPI has been chosen to parallelize the code in order to allow parallel computations both on a multi core laptop and large clusters. Parallelization of MSS, hence of SWIFT, is designed to fulfil two goals:

- Speedup of calculation,
- Capability to compute domains too large to fit in the memory of a single core

To achieve this, parallelization of SWIFT is twofold: geographical parallelization (GP) splits a domain in smaller tiles that do fit in a single core memory. Then time frame parallelization (TP) uses the diagnostic property of the code to compute

different time frames on different cores at the same time. Since no communications are needed between two time frames, speedup is very efficient under TP.

More specifically, SWIFT checks if there are enough cores to divide the domain in tiles (GP) according to the user request. Then with leftover cores, it checks if several time frames can run in parallel (TP). SWIFT can also be forced to do only TP. SWIFT needs also a master core to drive the calculation. The following picture shows an example of a 17 core calculation on an 8 tile domain that can compute simultaneously two time frames.

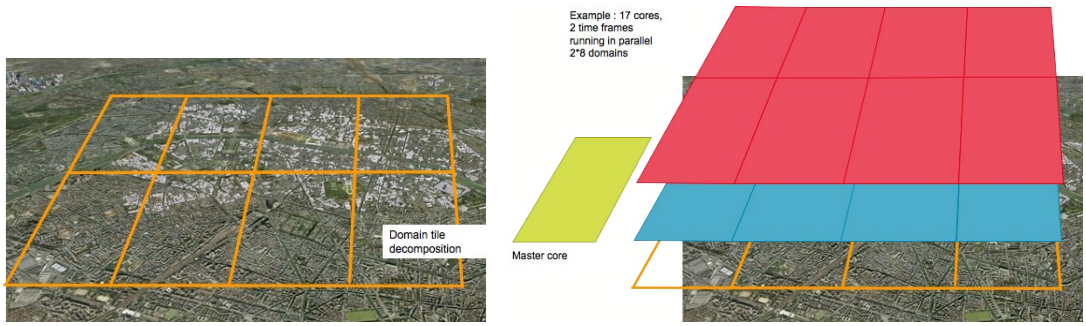


Figure 5: single domain parallelization example using 17 cores

The master core is in charge of this workload division. Its role is also to read the input data and distribute them to specific cores. Geographic information such as topography or roughness is divided into tiles and sent to the cores attached to each tile. Similarly, building data are transferred according to tile location. Not only are the buildings in the tile transferred, but also those from a safety zone around the tile: for instance, a building outside of the perimeter may have a wake that does intersect the tile. More over, wind interpolation needs conservative flux across tile boundaries for convergence of the mass consistent solver.

The master core does also the time slicing by sending correct time frames to specific cores: all meteorological data from a single time frame are transferred to each core, independently of their geographic location. In this way, there is smooth transition of interpolation across tile boundaries. Since the number of points where data are to be interpolated is less in a tile than in the whole domain, there is still a speedup even if the full set of meteorological measurements is transferred.

These parallel instructions are performed at the code top level. Parallel code modifications were designed to remain located on very specific parts of the code to preserve code clarity. The second set of parallel instructions is at the code deepest level of the adjustment solver. Since tiles are not overlapping, boundary quantities are therefore exchanged between cores handling neighbouring tiles. This is done:

- At the beginning of the time frame computation for static data such as topography, Eulerian 3D arrays that defines if cells are full or empty, initially interpolated wind, …
- During the computation, for computed data: wind correction but also specific convergence criteria,
- At the end of the computation for adjusted wind.

Computation stops if all tile convergence criteria are reached.

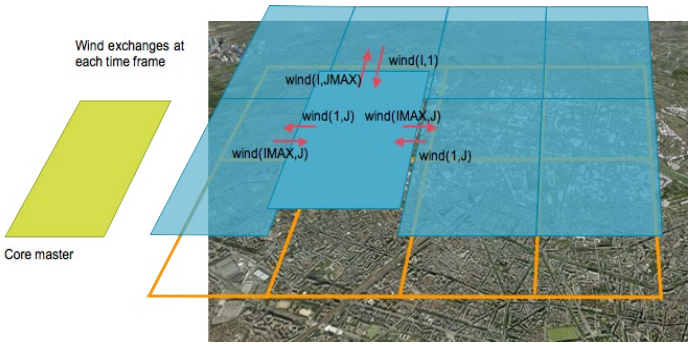


Figure 6: boundary data exchanges for a typical tile during adjustment step iteration

Figure 6 illustrates communications of a specific tile during an iteration of a time frame computation

SPRAY

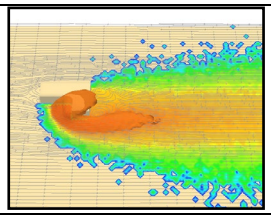
Dispersion is parallelized within Parallel SPRAY according to meteorological tile configuration received by SWIFT. If SWIFT provides only one tile to SPRAY, a single-tile parallelization is done. If SWIFT sends several meteorological tiles, a multiple-tile parallelization is performed.

Single-Tile Parallelization (STP)

If there is a single tile, SPRAY does not need to treat exchanges of mass between several adjacent tiles. The particles emitted by the sources involved in the simulation are simply distributed between available cores. Since the particles are randomly picked up from available sources, a natural balancing occurs between fast and slow particles. If no interactions between particles are needed, meaning that heavy gas computation is off, cores simply compute concentration from the subset of the total number of particles they are handling. Each time storage of concentration is required, the master calculates the summation of the 3D concentration fields provided by each core and writes the result into a binary file.

In order to respect the statistical properties of the model, each core random generator starts from a different random generator seed. Should the number of processors devoted to the computation be larger, the total number of emitted particles remains the same, but the history of individual particles end up being different: each random generator has its own history of numbers. Some parallelization discrepancies can appear, especially when the number of particles is small and close to the border of the plume.

The STP requires very limited MPI communications, these being broadcasting of input data by the master core at the beginning of the run and transmission of deposition 2D fields and concentration 3D fields at the end of every storage time step. So long as the storage time step is large compared to the displacement time step of particles, the speedup obtained by increasing the number of processors is quasi-linear (Figure 7). The slope of the speedup curve tends to decrease only when the number of particles moved by each individual core becomes too small compared to the load of MPI communication.



Cores	CPU (sec)	Speedup
2	274	1,00
3	138	1,99
4	91	3,01
5	69	3,97
6	56	4,89
7	47	5,85
8	41	6,77
15	18	14,89

Figure 7: EMU case – Single Tile speedup factor for PSPRAY

Multiple-Tiles Parallelization (MTP)

When SWIFT provides a multiple tile flow to SPRAY, SPRAY first locates sources that emit mass at the initial time of the computation. Each flow tile where an emitting source is found is activated. The cores involved in the computation are allocated to activated flow tiles, with respect to the mass rate of the sources inside them. If the mass emitted per second in a tile is twice the mass emitted per second in another tile, the number of cores affected to the first file will also be twice the number of cores affected to the second tile (see Figure 8).

In each activated flow-tile, the total number of particles emitted by the sources is distributed to cores, in a similar process as for the single-tile parallelization. A master core is elected for each tile: it broadcasts input data related to its flow-tile and receives 2D deposition/3D concentration fields each time a storage is required. Specific MPI communicators (SMPIC) are also defined: they group the set of cores working on the same tile.

Due to their own history, particles can cross the lateral boundaries of a tile: SPRAY performs the following steps, when particles of a tile, for instance tile A, where they were emitted, enter in another one, named tile B.

- If tile B was not activated, the model activates it and cores, taken from other tiles, are allocated to tile B,
- If Tile B was already activated, the group of particles is sent by tile A to tile B

Figure 9 illustrates this process.

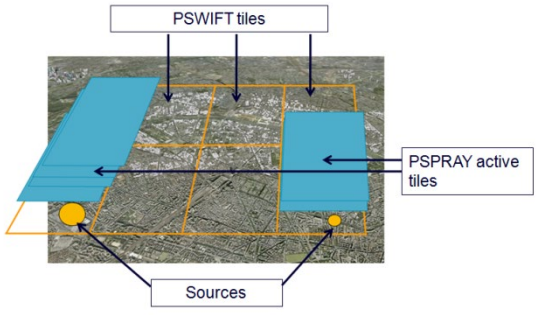


Figure 8: Initial distribution of cores done by LPD model

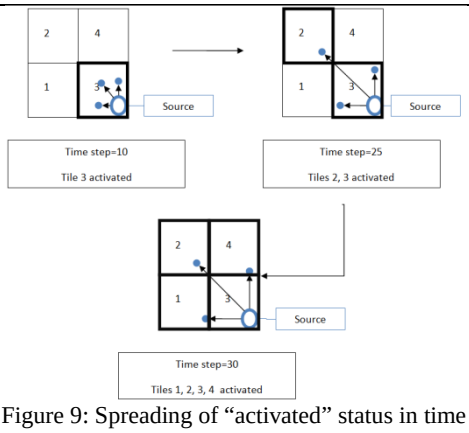


Figure 9: Spreading of "activated" status in time

In SPRAY model, the synchronisation time step of particles drives the frequency of exchange between cores. Its value varies typically from 1 second to 10 seconds. At the end of every synchronisation time step, every core produces packages containing particles leaving its tile (see Figure 10). If new tiles must be loaded or if a balancing process is required, a new allocation map of cores is calculated (Figure 11). Then the packages of particles are sent between cores (Figure 12). These packages are split according to the number of cores devoted to the tile they are travelling to: for instance, if particles are moving from tile A to tile B and if four cores are working on tile B, four groups of particles will be send to tile B.

Figure 10: Packages of particles are prepared

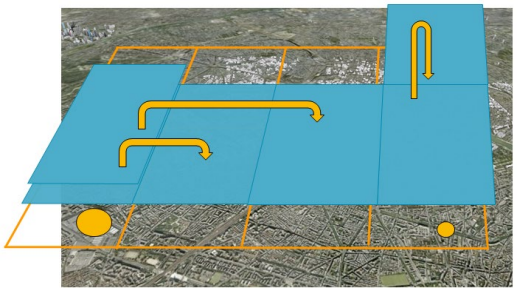


Figure 11: If new tiles are activated, a new allocation map of cores is computed

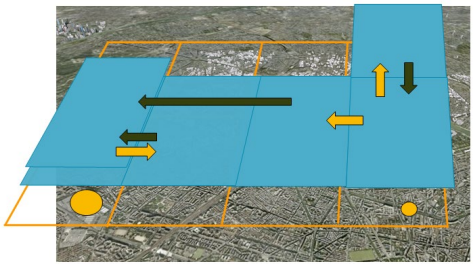


Figure 12: Exchange of particles between cores

Particle migrations between tiles occur very often. Cores therefore tend to end up rapidly with large differences in the amount of particles they are handling. In order to overcome this issue, a load balancing process is performed by SPRAY. The balancing process first computes the total number of particles located in the different tiles. Cores are added to overloaded tiles and removed from tiles with limited amount of particles. Then, inside each tile, a second balancing is performed: particles are equally distributed between cores devoted to the same tile. This two level load balancing process is automatically done by SPRAY when a new tile is activated or per request based on a user defined frequency. A typical value of this balancing time step is 60 seconds.

TEST CASES

Parallel MSS has been heavily tested on a full set of test cases, including:

- EMU, wind tunnel experiment of passive scalar transport and dispersion performed at Enflo Laboratory, Surrey University, UK,
- Mock Urban Setting Test (MUST) performed in 2001 at the West Desert Test Center, Utah,
- Joint Urban 2003 at Oklahoma City (OKC).

Regarding Parallel SWIFT, wind field differences between single tile and multiple tiles remain similar to round-off errors.

As far as speedup is concerned, Figure 13 shows the speedup factor for a TP of a 40 time frame run on MUST. Results are very good. Speedup factor drops predictably when reaching 40 cores since no more than 40 time frames can be run simultaneously.

Figure 13 also displays speedup results of a GP on OKC test case. Predictably, speedup factor is not as good as for a TP: GP purpose is memory constraints rather than speedup.

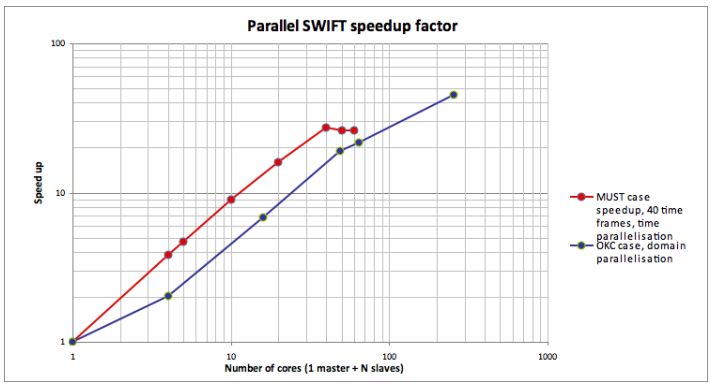


Figure 13: Parallel SWIFT speedup factor

Pspray

As far as PSPRAY is concerned, tests cases like OKC-IOP2 (Figure 15) and MUST have shown that (Figure 14):

- If the domain of study is small enough to be treated as a single tile, a STP provides a larger speedup factor than a MTP. Indeed, a MTP requires more MPI communications,
- The larger the number of emitted particles, the better the maximum speedup factor,
- Load balancing process is crucial to obtain a good speedup factor in MTP.

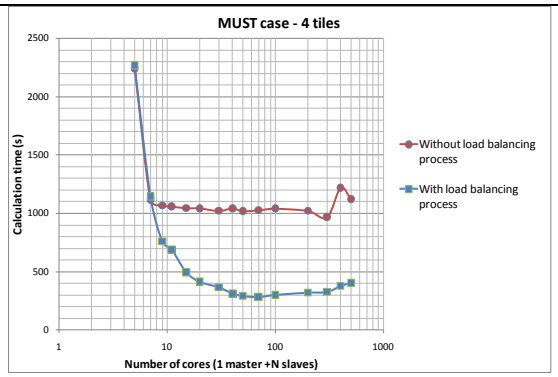


Figure 14: Dependence of calculation time on the number of cores and balancing process switched on or not

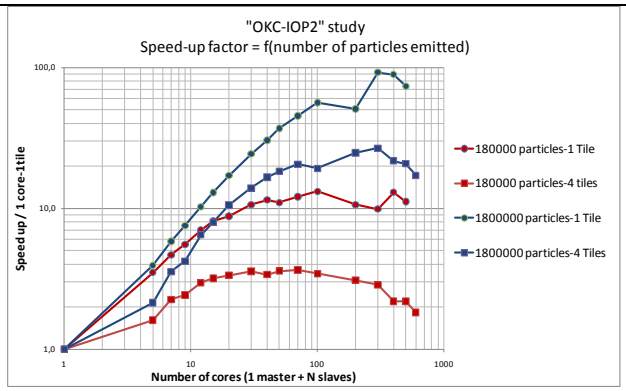


Figure 15: Speedup factor analysis on "OKC-IOP2" study

Analysis of numerous test cases shows that particles are smoothly exchanged between multiple tiles (see Figure 16). STP and MTP applied on the same case provide identical 3D concentration fields, except near the border of the plume. This is related to the statistical nature of the model and the random generators respectively used on 5 cores working on 4 tiles or on 1 core working on 1 tile.

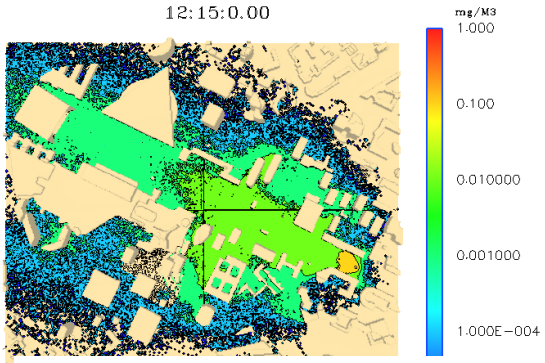


Figure 16: four-tile case on La Défense, the financial district west of Paris

CONCLUSION

Parallel MSS provides the same results quality as MSS and allows for computations over very large build-up areas. PMSS achieves very efficient speedup factor that makes possible its integration in an emergency response platform. Results over the whole city of Paris with a meso- to micro-scale meteorological forecast and hypothetical deleterious releases are presented in Duchenne *et al*, 2011.

Usage should be also extended to large city air pollution modelling through the AIRCITY project that plans to compute air pollution in cities with a spatial resolution of only a few meters.

REFERENCES

Anfossi D., F. Desiato, G. Tinarelli, G. Brusasca, E. Ferrero, D. Sacchetti (1998) "TRANSALP 1989 experimental campaignPart II: Simulation of a tracer experiment with Lagrangian particle models". *Atmospheric Environment*, **32**, 1157-1166

Carvalho J., D.Anfossi, S. Trini Castelli, G. A. Degrazia (2002) "Application of a model system for the study of transport and diffusion in complex terrain to the TRACT experiment". *Atmospheric Environment*, **36**, 1147-1161

Duchenne C., P. Armand, O. Oldrini, C Olry and J. Moussafir (2011), "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", *14th International Conference on Harmonisation within Atmospheric Dispersion Modelling fir Regulatory Purposes, Harmo'14, Kos (Greece), Oct. 2-6, 2011*

Ferrero E., D.Anfossi (1998) "Comparison of PDFs, closures schemes and turbulence parameterizations in Lagrangian Stochastic Models". *Int. J. Environment and Pollution*, Vol. 9, 384-410

Ferrero E., Anfossi D. and Tinarelli G. (2001) "Simulations of Atmospheric Dispersion in an Urban Stable Boundary Layer". *Int. J. Environment and Pollution*, **16**, 1-6

Glendening, J.W., J.A. Businger, and R.J. Farber, (1984), "Improving plume rise prediction accuracy for stable atmospheres with complex vertical structure". J. Air Pollut. Control Ass., 34 : 1128–1133

Kerr A., D. Anfossi, J. Carvalho, S. Trini Castelli (2001) "A dispersion study of the aerosol emitted by fertilizer plants in the region of Serra do Mar Sierra, Cubatão, Brazil". *Int. J. Environment and Pollution*, **16**, 251-263

Moussafir J., Oldrini O., Tinarelli G, Sontowski J, Dougherty C., (2004) A new operational approach to deal with dispersion around obstacles: the MSS (Micro-Swift-Spray) software suite. *Proc. 9th Int. Conf. on Harmonisation within Atmospheric Dispersion Modelling for Regulatory Purposes*, vol. 2, 114-118

Moussafir J., Buty D., Olry C., Nibart M., Albergel A., Tinarelli G., Anfossi D., Oldrini O., Harris T. and Dougherty C. (2007) SWIFT and MSS Current Developments. *11th Annual George Mason University Conference on Atmospheric Transport and Dispersion Modeling*. Fairfax, VA (USA), July 10-12, 2007

Thomson D.J. (1987) Criteria for the selection of stochastic models of particle trajectories in turbulent flows. *J. Fluid Mech*., Vol. 180, pp.529-556

Tinarelli G., Anfossi D., Brusasca G., Ferrero E., Giostra U., Morselli M.G., Moussafir J., Tampieri F. and Trombetti F., 1994: Lagrangian particle simulation of tracer dispersion in the lee of a schematic two-dimensional hill. Journal of Applied Meteorology, 33, N. 6, 744-756

Tinarelli G., Anfossi D., Bider M., Ferrero E.and Trini Castelli S. (2000) A new high performance version of the Lagrangian particle dispersion model SPRAY, some case studies. *Air Pollution Modelling and its Applications XIII*, S.E. Gryning and E. Batchvarova eds. Kluwer Academic / Plenum Press, New York, pp.499-507

Tinarelli G., Brusasca G., O.Oldrini, D.Anfossi, S.Trini Castelli, J.Moussafir (2007) "Micro-Swift-Spray (MSS) a new modelling system for the simulation of dispersion at microscale. General description and validation". Air Pollution Modelling and its Applications XVII, C. Borrego and A.N. Norman eds., Springer, 449-458

Trini Castelli S., Anfossi D., Ferrero E. (2003) "Evaluation of the environmental impact of two different heating scenarios in +urban area". *Int. J. Environment and Pollution*, **20**, 207-217