17th International Conference on
Harmonisation within Atmospheric Dispersion Modelling for Regulatory Purposes
9-12 May 2016, Budapest, Hungary

DEVELOPMENT OF THE PARALLEL VERSION OF A CFD – RANS FLOW MODEL ADAPTED TO THE FAST RESPONSE IN BUILT-UP ENVIRONMENTS

O. Oldrini[1], M. Nibart[2], P. Armand[3], J. Moussafir[2] and C. Duchenne[3]

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

Abstract:
Momentum SWIFT is a fast RANS flow model derived from Micro SWIFT. It uses artificial compressibility to solve steady state momentum equations. Rapidity of the momentum solver relies on simplified turbulence closure and regularity of the horizontal mesh. To be able to gain more speed and be able to handle larger size domains, the momentum solver has been parallelized.
Parallelization has been achieved using Message Passing Interface (MPI). Computation can be parallelized either in time, running multiple time frames on different cores, or in space, a large domain can be divided into multiple tiles and distributed to cores. Time and space parallelization schemes can be used at the same time. Momentum SWIFT can be used either on computer ranging from multi core laptop to large cluster. Space parallelization allows Momentum SWIFT to handle very large domains.
Validation has been performed against scalar version on multiple test cases. Test cases are ranging from academic isolated cube to complex city centers, like Michelstadt wind tunnel experiment, or Joint Urban 03 experiment on Oklahoma City. The latest has also been used in Udinee exercise. CPU performances of the model have also been derived both on multi core laptop and cluster.

Key words: *SWIFT, Micro-SWIFT, CFD RANS, parallel modelling**.***

INTRODUCTION

SWIFT (Oldrini et al., 2013) is a very fast mass consistent wind model. It produces mass consistent 3D wind fields from a set of sparse wind measurements or larger scale model outputs. Its range of application has been extended to local scale build-up areas using Röckle type parameterisation for buildings. SWIFT is referred as Micro-SWIFT (Moussafir et al. 2004) for such applications. Micro-SWIFT interpolates available meteorological data in 3D and creates analytical zones attached to various buildings, such as displacement, cavity, wake or skimming. Mass consistency is finally applied to obtain a divergence free wind field. Micro-SWIFT can be used in conjunction with the Lagrangian Particle Dispersion model SPRAY (Tinarelli et al., 2013, Tinarelli et al, 94). The modelling system is called MSS, Micro-SWIFT-SPRAY (Moussafir et al., 2004, Tinarelli et al., 2007), and is designed to model transport and dispersion of pollutants at local scale in build-up environment, ranging from industrial facilities to city centres..

SWIFT / Micro-SWIFT can also use momentum conservation (Oldrini et al. 2014). Simplified turbulence modelling and artificial compressibility scheme allow for fast calculation of steady state wind flow solutions. This capability can be used at the finer scale of a downscaling computation over a large city like Paris, around specific buildings of interest to compute infiltration of dangerous contaminants. Quality of wind field tends to be very similar to more general CFD codes, with very short computational time.

Parallelization scheme has been introduced into Momentum-SWIFT to allow for even faster calculations and also to extend the size of domain. After presenting the parallel implementation, results of parallel calculation are compared to scalar version and performances are evaluated.

PARALLEL SCHEME

Overview

The parallel scheme has been introduced consistently with the methodology described in Oldrini et al., 2011. Both weak and strong scaling have been implemented into Momentum-SWIFT:

- Weak scaling allows the code to handle arbitrary large problem by using domain decomposition (DD),
- Strong scaling is obtained by using the diagnostic property of the code: each time frame can be handled independently. This mode is referred later on as TFP (time frame parallelization).

DD can be used also to do strong scaling, but the communications are more numerous than in TFP, and the efficiency is hence less.

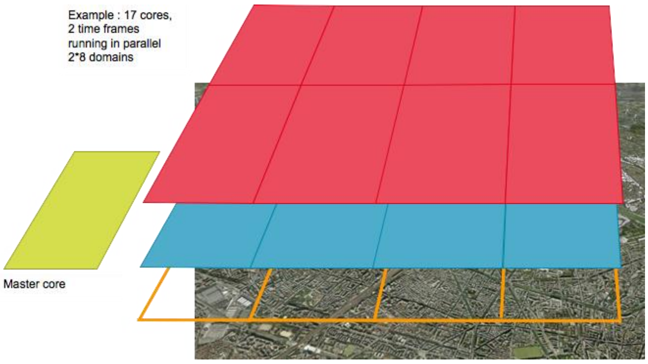

Figure 1. Example of TFP + DD for 17 cores, the domain being divided into 8 tiles, two time frames being computed in parallel

To be able to use the parallel scheme both on a laptop but also on a large cluster, Message Passing Interface (MPI) technology has been chosen.

The algorithm uses a master core that drives the computation and split the workload. The master core uses the number of cores available to distribute the workload to do DD, TFP or both. The master core accesses then all the input data, like topography or buildings, and is in charge to distribute them if DD is active. It also reads the meteorological data and distributes them if TFP is active: once a group of cores has finished computing a time frame, the master core sends them a new one.

TFP consists only in workload distribution on groups of cores by the core master: once the meteorological information is distributed, no additional communications are needed. Domain decomposition is more complex.

Domain decomposition

Using domain decomposition algorithm, the domain is divided in fixed size tiles. The user provides this size, chosen to fit in the memory of each single core.

When using DD, communications are obliviously necessary at the code top level. The master core handles the input data: topography, land use, roughness, buildings, etc. The master core splits this information and distributes it to each core according to the tile he is working on.

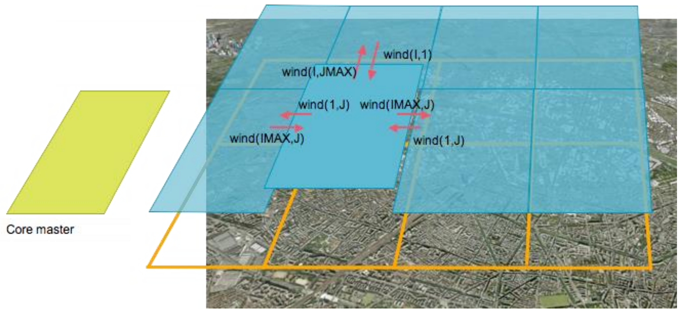

Figure 2. Boundary exchanges for wind field calculation in case of domain decomposition in 8 tiles

Then, at the solver level, communications are also necessary for the artificial compressibility scheme that solves mass and momentum equations. Derivatives are computed to solve the equations, and for grid points at tile boundaries, information is needed from neighbouring tiles. The figure 2 displays wind information exchanges needed during each artificial compressibility step. In the same way, pressure is also exchanged but at each algorithm sub step (pressure projection).

COMPARISON WITH THE SCALAR VERSION OF THE CODE

Parallel-Momentum-SWIFT has been tested against the scalar version.

The test cases chosen are:

- Academic cube,
- JU2003,
- Michelstadt.

The academic cube is a 20m cubic obstacle inside a 190 x 160m domain, with a metric resolution. The vertical grid has 13 nodes going up to 90m.

JU2003 (Allwine et al., 2004) is a field experiment conducted in Oklahoma City (OKC). Our test case is focusing on downtown measurements. The domain has 251 x 251 x 26 grid points and is covering 1km x 1km x 400m in OKC city centre.

Michelstadt (Leitl et al., 2014) is a wind tunnel experiment performed at Hamburg wind tunnel. It reproduced an idealized European city centre. The domain has 533 x 311 x 26 grid points and is covering 1.6km x 930m x 200m.

The following pictures are displaying a slice of the wind field produced by the scalar version and the parallel version seen from above, and with a domain split in four tiles (Cube / JU2003) and 6 tiles (Michelstadt).

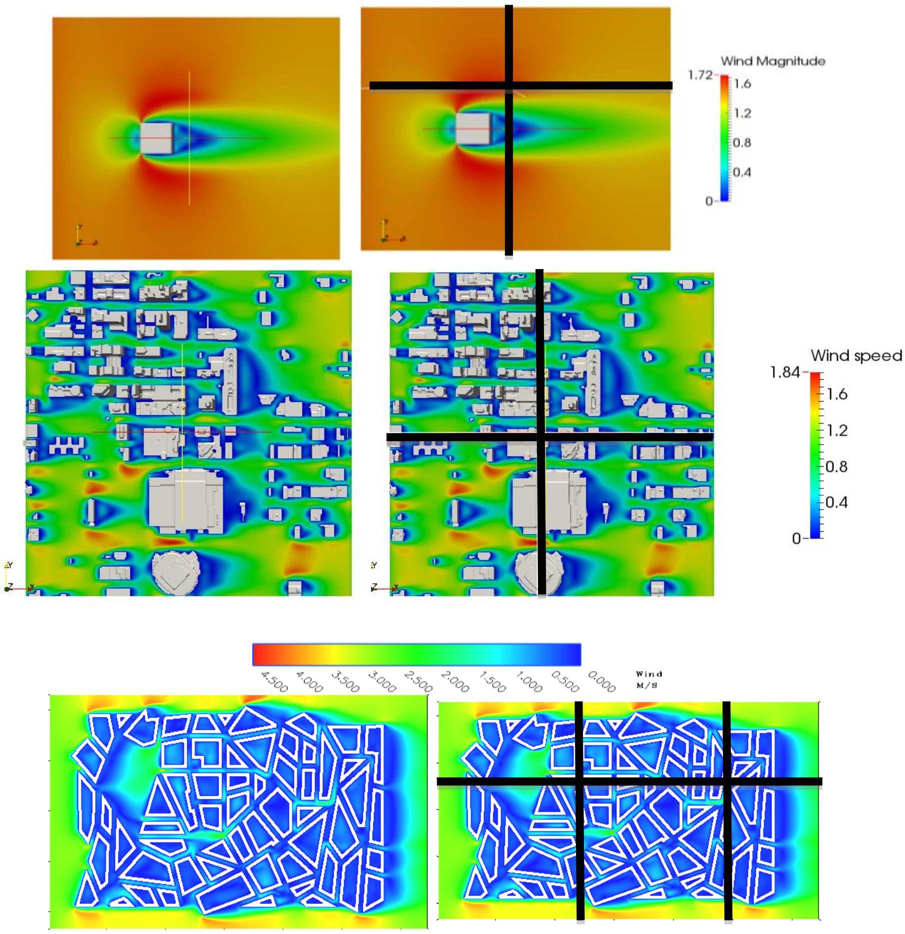

Figure 3. Top view of wind intensity. The scalar version is on the left side, the parallel version on the right side. The top picture is the academic cube (slice at 10m above ground), the one in the middle is JU2003 (slice at 1m above ground) and the bottom one is Michelstadt (slice at 6m above ground)

The differences for the wind field are below 5% between the scalar version and the parallel one.

EFFICIENCY OF DOMAIN DECOMPOSITION

The speedup S_n is calculated as the ratio of the duration of the calculation using a single core, T_1, divided by the duration of the calculation on n cores, T_n.

$$S_n = T_1 / T_n$$

S_n is ideal when $S_n = n$, which means that the calculation is n time fasters when using n cores. The speedup is limited due to cost of communications, but also due to the fraction of the model that has been parallelized (Amdahl's law).

Preliminary tests on efficiency have been performed on the previous validation test cases.

The academic cube and the OKC grids are small grids and speedup has been evaluated on a multicore laptop. DD with four tiles has been compared to single tile calculation.

The laptop used is a dual core (two processors) but with hyper threading technology, which should increase the physical processors to 4 logical processors.

Table 1. Speedup obtained on small test cases (cube and JU2003)

Test case	Academic cube	JU2003
Number of cores	2 physical cores, 4 using hyper threading	
Speedup	1.5	2
Ideal speedup	2-4	2-4

The speedup is acceptable, especially considering the grid size: the test cases are small and DD leads to less than 100 x 100 x 23 grid points for each tile. The workload is small and the parallel fraction of the code is hence limited.

When testing DD on Michelstadt test case, the performances have been evaluated on two nodes of a cluster, each node consisting in 12 processors. The DD uses 6, 9 and 16 tiles and the speedup obtained is presented in table below.

Table 2. speedup obtained on Michelstadt case using DD

Test case	Michelstadt		
Number of cores	6	9	16
Speedup	2	4	6
Ideal speedup	6	9	16

The efficiency, defined as the ratio of the actual speedup to the ideal speedup, is around 30-40%. This efficiency is quite common when doing parallel computing and is compatible with the primal objective of DD: DD was implemented for weak scaling, i.e. being able to compute very large domains, too large to fit in the memory of a single computer. Nonetheless overall performances in a strong scaling view are satisfying.

CONCLUSION

Parallel algorithms have been introduced into Momentum-SWIFT. These algorithms, domain decomposition and time frame parallelization, make it possible to increase the rapidity of the code (strong scaling), but also to compute very large domains (weak scaling).

Comparison on several tests cases, academic, wind tunnel and field experiment, are good. Scalar calculations and parallel calculations are presenting differences below 5% for wind field.

Regarding parallel performances, preliminary DD tests have been performed on the same test cases. DD efficiency is satisfying, even for small domains. More tests are being realized on more intensive test cases, like very large domains.

REFERENCES

Allwine, K. J., Leach, M. J., Stockham, L. W., Shinn, J. S., Hosker, R. P., Bowers, J. F., & Pace, J. C. , 2004. Overview of Joint Urban 2003—an atmospheric dispersion study in Oklahoma City. In Preprints, Symp. on Planning, Nowcasting and Forecasting in the Urban Zone, Seattle, WA, Amer. Meteor. Soc., CD-ROM J (Vol. 7).

Archambeau, F., Mechitoua, N., Sakiz, M., 2004: Code Saturne: a finite volume method for the computation of turbulent incompressible flows, *International Journal of Finite Volumes*.

Gowardhan A. A., E. R. Pardyjak, I. Senocak, M. J. Brown, 2011: A CFD-based wind solver for urban response transport and dispersion model, *Env. Fluid Mech.*, 11:439-464

Leitl, B., Castelli, S. T., Baumann-Stanzer, K., Reisin, T. G., Barmpas, P., Balczo, M., ... & Milliez, M., 2014. Evaluation of Air Pollution Models for Their Use in Emergency Response Tools in Built Environments: The 'Michelstadt'Case Study in COST ES1006 ACTION. In Air Pollution Modeling and its Application XXIII (pp. 395-399). Springer International Publishing.

Moussafir J., C. Olry, M. Nibart, A. Albergel, P. Armand, C. Duchenne, F. Mahe, L. Thobois and O. Oldrini, 2013: Aircity, a very High-resolution 3D Atmospheric Dispersion Modeling System for Paris. 15th Int. Conf. on Harmonisation within Atmospheric Dispersion Modelling for Regulatory Purposes

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

Napoly A., 2013, Simulation numérique 3D de la dispersion de rejets atmosphériques accidentels. Comparaison d'une approche détaillée et simplifiée de modélisation, Training report.

Nibart M., Armand P., Olry C., Duchenne C. and Albergel A., 2011, The indoor / outdoor pollutant transfer of a hazardous release: application to a parisian railway station, proc. 14th int. Conf. on Harmonisation within Atmospheric Dispersion Modelling for Regulatory Purposes

Oldrini, O., Nibart, M., Armand, P., Olry, C., Moussafir, J., & Albergel, A., Introduction of Momentum Equations in Micro-SWIFT, 16th International Conference of Harmonization within Atmospheric Dispersion Modeling for Regulatory Purposes. September 2014, Varna

Oldrini, O., Nibart, M., Armand, P., Olry, C., Moussafir, J., & Albergel, A., Multi Scale Built-up Area Integration in Parallel SWIFT, 15th International Conference on Harmonization within Atmospheric Dispersion Modeling for Regulatory Purposes. May 2013, Madrid

Oldrini O., C. Olry, J. Moussafir, P. Armand and C. Duchenne, 2011: Development of PMSS, the Parallel Version of Micro SWIFT SPRAY. Proc. 14th Int. Conf. on Harmonisation within Atmospheric Dispersion Modelling for Regulatory Purposes, 443-447

Tinarelli G., Mortarini L., Castelli S.T., Carlino G., Moussafir J., Olry C., Armand P., Anfossi D., 2013: Review and validation of MicroSpray, a lagrangian particle model of turbulent dispersion, *Journal of Geophysical Research,* **200,** 311-327.

Tinarelli G., G. Brusasca, 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