

Adaptation of Multidimensional Positive Definite Advection Transport Algorithm to Modern High-Performance Computing Platforms

Bogdan Rosa, Lukasz Szustak, Andrzej A. Wyszogrodzki, Krzysztof Rojek, Damian K. Wójcik, and Roman Wyrzykowski

Abstract—In this study, we report development of two innovatory parallel implementations of the Multidimensional Positive Definite Advection Transport Algorithm (MPDATA). MPDATA is the main module of the multiscale fluid model EULAG [Prusa *et al.*, *Computers & Fluids*, vol. 37, 2008]. Recently, the dynamical core of EULAG has been implemented into COSMO (Consortium for Small-scale Modeling) weather prediction framework and is expected to be in operational use.

The original code of MPDATA is written in FORTRAN77 and has excellent efficiency and scalability on conventional supercomputer architectures. The new C++ implementations are designed and optimized under modern CPU and GPU based high-performance computing platforms. A number of innovatory solutions have been employed in these codes, including stencil decomposition, block decomposition (with weighting analysis between computation and communication), reduction of intercache communication by partitioning of cores into independent teams, cache reusing and vectorization. The new techniques allow to take the full advantage of the new architectures and accelerate the code execution.

The correctness and accuracy of the new implementations are examined based on a standard three-dimensional solid body rotation test case. Additionally, we focus on testing computational efficiency and scalability. In most runs and especially in simulations with larger computational grids the new codes performed better than the traditional implementation.

Index Terms—MPDATA, advection solver, parallel computing.

I. INTRODUCTION

In recent years, there has been a rapid increase in using modern supercomputing architectures for modeling complex engineering systems. An important example of such applications is numerical weather prediction (NWP). In NWP the physical processes governing atmospheric flows are simulated by solving partial differential equations in three-dimensional space and time. Initial and boundary conditions are provided by a forecast of the global model, run at the coarse grid resolution. Computational complexity of

modeling mesoscale weather systems results from a wide range of scales involved. Spatial scales of atmospheric flows range from global (10^7 m) to micro (about 1 m) for the boundary layer turbulence and even much less (10^{-5} m) for microphysical processes in clouds. This requires, possibly high spatial resolution to adequately resolve all essential physical processes affecting the weather systems. On the other hand a solution time has to be significantly shorter than the time of physical realization of the process itself. Therefore, it is necessary to employ efficient numerical algorithms and large computational resources. The new hybrid and heterogeneous supercomputers based on CPU/GPU architectures offers unique opportunities for modeling atmospheric processes significantly faster and with accuracy greater than ever before. However, the traditional codes are bottlenecked by cache performance and memory bandwidth and therefore do not allow to take full advantage of the new architectures. To be able to effectively exploit the potential of new hybrid platforms the structure of the traditional codes must be significantly redesigned.

In this study we focus on adapting Multidimensional Positive Definite Advection Transport Algorithm [1]-[5] to GPU-based architectures such as GeForce GTX TITAN and new CPU (Central Processing Unit) e.g. Intel Xeon E5-2695. MPDATA is the main module of the multiscale fluid model EULAG [6]. The model is an innovative solver in the field of numerical modeling of multiscale atmospheric flows. MPDATA solves continuity equation describing the advection of a nondiffusive quantity ψ in a flow field, namely

$$\frac{\partial \psi}{\partial t} + \text{div}(\mathbf{V}\psi) = 0, \quad (1)$$

where \mathbf{V} is the velocity vector. The algorithm is positive defined and by appropriate flux correction [4] can be also monotonic. This is a desirable feature for advection of positive definite variables such as specific humidity, cloud water, cloud ice, rain, snow, aerosol particles and gaseous substances. The spatial discretization of MPDATA is based on finite difference approximations. The algorithm is iterative and fast convergent. In the first sub-step, advection of a prognostic field ψ is computed with the standard donor-cell approximation [1]. This ensures the first order of accuracy only. In the subsequent time step corrections are applied to make the scheme more accurate i.e. second order in space and time. In the corrective sub-step the donor-cell approximation is used again but with new anti-diffusive

Manuscript received March 15, 2015; revised May 19, 2015. This work was supported by the Polish National Science Centre under grant no. UMO-2011/03/B/ST6/03500.

B. Rosa, A. A. Wyszogrodzki, and D. K. Wójcik are with the Institute of Meteorology and Water Management - National Research Institute, Podleśna 61 Street, 01-673 Warsaw, Poland (e-mail: bogdan.rosa@imgw.pl).

L. Szustak, K. Rojek, and R. Wyrzykowski are with the Czestochowa University of Technology, Dabrowskiego 69 Street, 42-201 Czestochowa, Poland.

velocities computed based on the advected fields. The procedure can be repeated many times, however typically after more than 2 corrections no significant improvements are observed.

Recently, the dynamical core of EULAG has been implemented into COSMO [7] weather prediction framework. The compressible dynamical core of the COSMO model uses two different schemes for the advection of tracers. These are semi-Lagrangian and direction-splitted finite volume (Bott-scheme). The semi-Lagrangian scheme is a fully 3-dimensional but does not conserve mass. The Bott-scheme allows for relatively large time steps but it is one-dimensional. In this context MPDATA have significant advantages, namely, is mass conserving and fully 3 dimensional. This ensures that the computational grid box will not be emptied in a single time step.

II. NEW C++ IMPLEMENTATION OF MPDATA FOR CPU AND GPU

The original code of MPDATA was written in FORTRAN77 and parallelized using MPI (Message Passing Interface) library. The parallelization strategy of the original implementation is based on 2D domain decomposition. The improved implementation [8] extends decomposition to three spatial directions (3D). The code has excellent efficiency and scalability on conventional supercomputer architectures. Recently we have developed two new C++ implementations, which are optimized under modern CPU and GPU supercomputers. The new codes allow more efficiently utilize computational resources. The structure of the algorithm has been significantly remodeled. With the new strategies of parallelization the computations can be evenly distributed across the entire machine.

In GPU implementation the main importance was attached to the optimization of the memory usage. Several different strategies for accelerating the code execution, such as space and temporal blocking or stencil decomposition [9] were tested, verified and compared. Eventually a modified technique of 2.5D spatial blocking [10] has been employed. This approach allows to increase data locality and minimize the number of global memory transaction. Detailed description of the new GPU implementation one can find in [11]-[14].

The new CPU implementation [15], [16] makes use of the (3+1)D decomposition strategy for the stencils computation. This approach allows to release memory more efficiently and hence helps to overcome the communication bounds. The main difficulty was to transfer the data traffic from main memory to cache hierarchy by proper reusing of the cache memory. This goal was achieved by employing a combination of the loop tiling and the loop fusion. Such optimizations cost some extra computations but in general provide benefit. Additionally, to improve the computational efficiency the algorithm groups the cores (threads) into independent work teams. It reduces inter-cache communication overheads due to less expensive communications between neighboring threads/cores, and direct synchronizations. Similarly to GPU implementation the new method allows to distribute computational tasks on

the available resources more efficiently.

Here, it is worth mentioning about alternative C++ implementations of MPDATA. Recently Jaruga *et al.* [17] developed C++ code of MPDATA making use of the Blitz++ library. Their implementation is based on domain decomposition and shared-memory parallelization.

III. SOLID BODY ROTATION TEST

The new (CPU and GPU) implementations of MPDATA are examined using the standard three-dimensional solid-body rotation test, in which a sphere (of radius R) is advected with constant angular velocity ($\omega=0.1$) around the domain diagonal axis. The velocity field is stationary and divergent free. The initial distribution of the tracer (density) inside the sphere is defined as $\psi(r) = 4(1 - r/R)$, where r is the distance from the center of the sphere. The initial concentration of ψ is shown in Fig. 1. The red isosurface represents the constant magnitude of tracer at 0.5. Distribution of ψ inside the sphere is visualized in the horizontal cross-section at the top of the sphere. The blue streamlines represent the flow pattern in the rotating fluid.

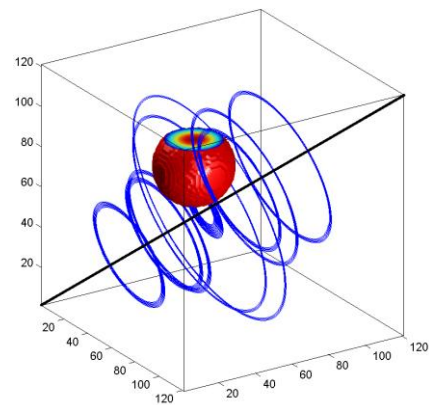


Fig. 1. Tracer concentration and streamlines at initial stage ($t = 0$).

Numerical simulations have been performed at regular grid ($\Delta x = \Delta y = \Delta z$) and three different resolutions, namely 41^3 , 81^3 , 121^3 . The time step size depends on the grid resolution. Assuming that the grid spacing is equal 2 and time increment $\Delta t = 0.2$ we estimate the maximum value of the Courant number at CFL=0.29.

Fig. 2 shows sequence of 3D tracer distribution after one full rotation. Analogously to Fig. 1, the red isosurfaces represent the constant magnitude of ψ at 0.5. The plots in rows correspond to three different grid resolutions, namely 41^3 , 81^3 and 121^3 . The second column shows results from simulations with the new CPU implementation. Correspondingly, solutions obtained with the GPU implementation are in column 3. Several important observations can be made from Fig. 2. First, there is qualitative agreement between solutions obtained with the new CPU and GPU implementations. Second, due to implicit numerical diffusion the surfaces of the spheres become smoother. Third, diffusion is not the same in every direction. Its intensity depends on the direction of rotation. This effect can be observed on the sphere's surface. The right lower side

of the sphere is smoother than its upper back side.

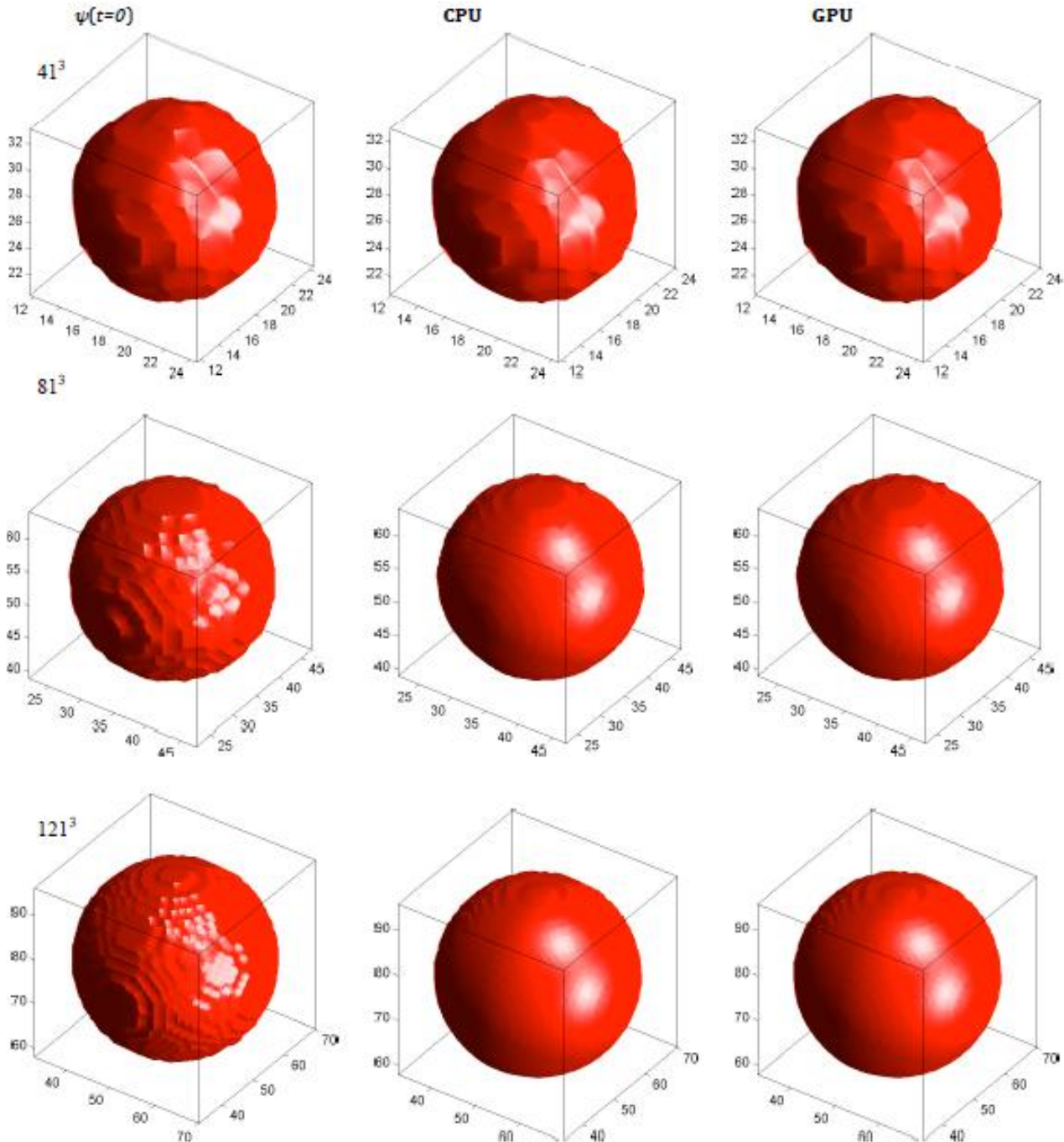


Fig. 2. Tracer concentrations at $t = 0$ (first column) and after one full rotation (columns 2-3).

A quantitative comparison of the numerical results is presented in Fig. 3. The sequence of three plots shows 2D horizontal cross-sections through center of the sphere. The thin solid line represents the initial location of the sphere. Thick solid lines indicate the location of the sphere after one full rotation. Solutions obtained with CPU and GPU

To quantify the numerical results we use the following statistical measures:

- 1) the phase error defined as the distance between the exact maximum position (at $t=0$) and computed one after one full rotation

$$E_{phase} = \sqrt{(i^{start} - i^{end})^2 + (j^{start} - j^{end})^2 + (k^{start} - k^{end})^2} \quad (2)$$

implementations are plotted using the same color table and contour levels. The Fig. 3 confirms that both solutions are in quantitative agreement. Results obtained with different implementations (CPU and GPU) agree to within 13 significant digits (see Table II) therefore the thick solid lines in Fig. 3 perfectly overlap.

- 2) The diffusion error

$$E_{diffusion} = \max(q^{start}) - \max(q^{end}), \quad (3)$$

- 3) And L2-norm

$$E_{L2} = \sqrt{\frac{1}{N} \sum_{i=1}^N (q_i^{start} - q_i^{end})^2}. \quad (4)$$

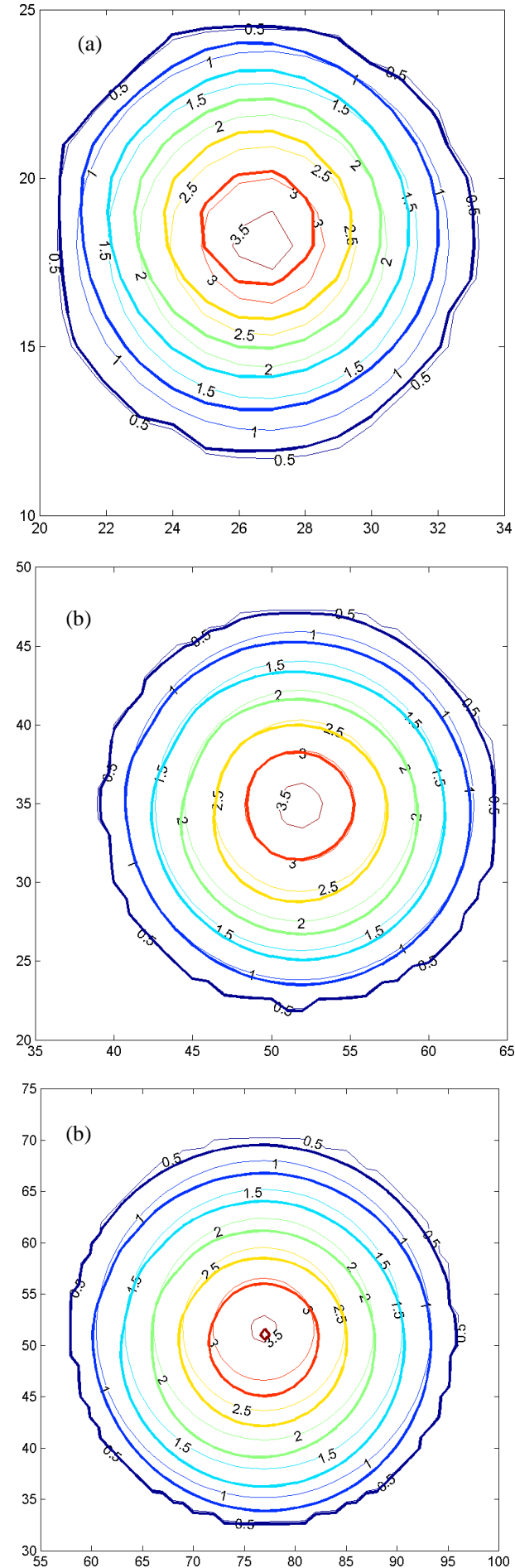


Fig. 3. Tracer concentrations in horizontal plane at $t = 0$ and after one full rotation. Three panels correspond to three different resolutions of the computational grid, namely (a) 41^3 , (b) 81^3 , (c) 121^3 .

Numerical values of these measures computed based on solution obtained with the new CPU implementation are listed in Table I. In line with expectations, the implicit numerical diffusion decreases with increasing mesh resolution. Both maximal value and diffusion decrease monotonically with the grid size. The only exception is $L2$. The value computed for 121^3 is larger than this for 81^3 . The deviation may result from the larger number of grid points and in consequence larger cumulative error.

TABLE I: BASIC STATISTICAL MEASURES TO QUANTIFY CPU RESULTS

Mesh size	$L2$ - norm E_{L2}	Maximum value	E_{phase}	$E_{diffusion}$
41^3	0.0324	3.198	$< \Delta x$	0.616
81^3	0.0221	3.475	$< \Delta x$	0.451
121^3	0.0314	3.583	$\sim \Delta x$	0.310

The differences between the statistical measures computed based on data obtained with different implementations i.e. CPU and GPU are listed in Table II. The data confirm quantitative agreement between simulations performed with the new codes. At grids 81^3 and 121^3 the accuracy reaches 15 significant digits that is close to machine precision.

TABLE II: THE DIFFERENCE BETWEEN CPU AND GPU SOLUTIONS (XCPU-XGPU)

Mesh size	$L2$ - norm	Maximum	
	E_{L2}	Value	$E_{diffusion}$
41^3	$\sim 10^{-11}$	$\sim 10^{-13}$	$\sim 10^{-13}$
81^3	$\sim 10^{-15}$	$< 10^{-16}$	$< 10^{-16}$
121^3	$< 10^{-16}$	$< 10^{-16}$	$\sim 10^{-16}$

IV. PARALLEL PERFORMANCE OF THE NEW IMPLEMENTATIONS

Parallel performance of the new CPU implementation has been examined using $2 \times$ Intel Xeon E5-2695 v2 2.4GHz (2x12cores). The testing runs were limited to four different computational meshes. Three of them were identical (the same size and dimensions) as those in the experiment with the rotating sphere. Two additional simulations were performed at larger grid ($1024 \times 512 \times 64$) and different number of time steps. Size and dimension of the larger grid is characteristic for the contemporary NWP simulations. Analogous set of tests has been performed for the GPU implementation. Parallel efficiency of the GPU code was examined using NVIDIA GeForce GTX TITAN (2688 CUDA Cores). We should note that in GPU tests we did not include the time for data transfer between RAM and GPU. The performance results are listed in Table III. For comparison, we listed also reference results obtained with the traditional CPU implementation.

TABLE III: PARALLEL PERFORMANCE. WALL CLOCK TIME [S]

Mesh size	# time steps	CPU	CPU-new	GPU
$41 \times 41 \times 41$	314	0.12	0.11	0.29
$81 \times 81 \times 81$	628	1.62	1.21	1.94
$121 \times 121 \times 121$	942	12.66	6.07	5.61
$1024 \times 512 \times 64$	157	52.92	18.14	14.28
$1024 \times 512 \times 64$	403	133.58	46.12	35.70

Table IV shows speedup of the new C++ codes. Here, the speedup is defined as a ratio of wall clock time obtained in runs with the traditional CPU implementation written in FORTRAN77 to the wall clock time obtained with the new C++ implementation. Several conclusions can be drawn by studying the results in Table IV. The new implementations perform better than the traditional FORTRAN77 CPU implementation in simulations of larger complexity. GPU code is not profitable for meshes smaller than $\sim 100^3$. For the most complex simulation the GPU implementation is 3.74 times more efficient than the original CPU implementation and about 20% more efficient than the new CPU implementation.

TABLE IV: PARALLEL PERFORMANCE SPEEDUP

Mesh size	# time steps	$T_{\text{CPU}}/T_{\text{CPU-new}}$	$T_{\text{CPU}}/T_{\text{GPU}}$	$T_{\text{CPU-new}}/T_{\text{GPU}}$
41×41×41	314	1.09	0.41	0.38
81×81×81	628	1.34	0.83	0.62
121×121×121	942	2.08	2.26	1.08
1024×512×64	157	2.92	3.71	1.27
1024×512×64	403	2.90	3.74	1.29

V. CONCLUSION

Two new C++ implementations of MPDATA have been developed. The new codes are optimized under modern CPU and GPU computing platforms. The parallelization strategy employed in the GPU code is based on modified 2.5D spatial blocking technique. This approach allows for more efficient load balancing and minimizes data communication. The new CPU implementation makes use of the innovatory (3+1)D decomposition strategy for the stencils computation.

Using the new codes we performed the standard three-dimensional solid-body rotation test. Numerical results show that there is quantitative agreement between solutions obtained with the new codes. In a series of experiments we analyzed numerical diffusion, phase shift and standard $L2$ norms. Simulations performed at different mesh resolution are used to show scalability, speedup, and overall efficiency. We conclude that the new codes perform better than original (FORTRAN77) implementation.

Adaptation of MPDATA to modern computing architectures is a step forward in improving performance of the whole EULAG model. This in turn results in reduction of the computation time and enable the use of high-resolution computational grids. The spatial resolution has direct impact on the accuracy of modeled weather phenomena. In particular, the high resolution is necessary to predict violent weather phenomena such as tornadoes, convective storms, heavy rain or wind shear. An expected benefit that may be obtained from the new implementation is improving the quality of and verifiability of weather forecasts.

REFERENCES

[1] P. K. Smolarkiewicz, "A simple positive definite advection scheme with small implicit diffusion," *Mon. Wea. Rev.*, vol. 111, no. 3, pp. 479-486, March 1983.

[2] P. K. Smolarkiewicz, "A fully multidimensional positive definite advection transport algorithm with small implicit diffusion," *J. Comp. Phys.*, vol. 54, no. 2, pp. 325-362, May 1984.

[3] P. K. Smolarkiewicz and L. G. Margolin, "MPDATA: A finite-difference solver for geophysical flows," *J. Comp. Phys.*, vol. 140, no. 2, pp. 459-480, March 1998.

[4] P. K. Smolarkiewicz and W. W. Grabowski, "The multidimensional positive definite advection transport algorithm: Nonoscillatory option," *J. Comp. Phys.*, vol. 86, no. 2, pp. 355-375, February 1990.

[5] P. K. Smolarkiewicz, "Multidimensional positive definite advection transport algorithm: An overview," *Int. J. Numer. Meth. Fluids.*, vol. 50, no. 10, pp. 1123-1144, April 2006.

[6] J. M. Prusa, P. K. Smolarkiewicz, and A. A. Wyszogrodzki, "EULAG, a computational model for multiscale flows," *Comp. Fluids*, vol. 139, no. 9, pp. 1193-1207, October 2008.

[7] M. Baldauf, A. Seifert, J. Forstner, D. Majewski, and M. Raschendorfer, "Operational convective-scale numerical weather prediction with the COSMO model: Description and sensitivities," *Mon. Wea. Rev.*, vol. 139, no. 12, pp. 3887-3905, December 2011.

[8] Z. P. Piotrowski, A. A. Wyszogrodzki, and P. K. Smolarkiewicz, "Towards petascale simulation of atmospheric circulations with soundproof equations," *Acta Geophys.*, vol. 59, no. 6, pp. 1294-1311, December 2011.

[9] A. Schafer and D. Fey, "High performance stencil code algorithms for GPGPUs," *Computer Science*, vol. 4, pp. 2027-2036, May 2011.

[10] A. Nguyen, N. Satish, J. Chhugani, K. Changkyu, and P. Dubey, "3.5-D blocking optimization for stencil computations on modern CPUs and GPUs," in *Proc. ACM/IEEE Int. Conf. for High Performance Computing, Networking, Storage and Analysis*, 2010, pp. 1-13.

[11] R. Wyrzykowski, L. Szustak, K. Rojek, and A. Tomas, "Towards efficient decomposition and parallelization of MPDATA on hybrid CPU-GPU cluster," *Lect. Notes in Comp. Sci.*, vol. 8353, pp. 434-444, June 2014.

[12] K. Rojek, L. Szustak, and R. Wyrzykowski, "Performance analysis for stencil-based 3D MPDATA algorithm on GPU architecture," *Lect. Notes in Comp. Sci.*, vol. 8384, pp. 145-154, May 2014.

[13] B. Rosa, M. Ciznicki, K. Rojek, D. K. Wójcik, P. K. Smolarkiewicz, and R. Wyrzykowski, "Porting multiscale fluid model EULAG to modern heterogeneous architectures," *International Journal of Applied Physics and Mathematics*, vol. 4, no. 3, pp. 188-195, May 2014.

[14] K. Rojek, M. Ciznicki, B. Rosa, P. Kopta, M. Kulczewski, K. Kurowski, Z. P. Piotrowski, L. Szustak, D. K. Wójcik, and R. Wyrzykowski, "Adaptation of fluid model EULAG to graphics processing unit architecture," *Concurrency and Computations: Practice and Experience*, vol. 27, no. 4, pp. 937-957, March 2015.

[15] L. Szustak, K. Rojek, T. Olas, L. Kuczynski, K. Halbiniak, and P. Gepner, "Adaptation of MPDATA heterogeneous stencil computation to Intel Xeon Phi coprocessor," *Scientific Programming*, DOI: 10.3233/SPR-140403.

[16] L. Szustak, K. Rojek, and P. Gepner, "Using Intel Xeon Phi coprocessor to accelerate computations in MPDATA algorithm," *Lect. Notes in Comp. Sci.*, vol. 8384, pp. 582-592, May 2014.

[17] A. Jaruga, S. Arabas, D. Jarecka, H. Pawlowska, P. K. Smolarkiewicz, and M. Waruszewski, "libmpdata++ 1.0: a library of parallel MPDATA solvers for systems of generalized transport equations," *Geosci. Model Dev. Discuss.*, vol. 8, pp. 1005-1032, April 2015.



Bogdan Rosa received his M.Sc. degree in physics from the University of Warsaw in 2000, followed by the Ph.D. in 2005. Afterwards, he spent 3 years as a postdoctoral fellow at the University of Delaware, Department of Mechanical Engineering, where he was involved in developing computational tools to study collision rates and growth of droplets in atmospheric clouds. Since 2009, Prof. Rosa is working at the Institute of Meteorology and Water Management - National Research Institute. His current projects include adaptation of the numerical model EULAG into a German/Swiss weather prediction model COSMO.



Lukasz Szustak received his M.Sc. in computer science from the Czestochowa University of Technology in 2008 and his PhD in 2012. During this period, his doctoral research focused on adaptation of high performance computing to modern parallel architectures including hybrid platforms. Since 2012, Dr. Szustak is employed at Czestochowa University of Technology. His current work is associated with the development of efficient methods of scheduling, load balancing, and adaptations of stencil based computations to Intel MIC and CPUs architectures.



Andrzej A. Wyszogrodzki received M.Sc. and Ph.D. degrees from the University of Warsaw, Poland, in physics in 1996 and 2001, respectively. Afterwards, he spent 3 years as a postdoctoral fellow at the Los Alamos National Laboratory in New Mexico, USA and one year at the IBM T. J. Watson Research Center, Yorktown Heights, USA. In 2005 he went to Boulder, Colorado, to work as a project scientist at the National Center for Atmospheric Research (NCAR). In September 2013 he

joined the Institute of Meteorology and Water Management - National Research Institute in Warsaw, Poland as a head of the department of the Numerical Weather Forecasting with COSMO model.



Damian K. Wójcik received M.Sc. degree in computer science in 2009 and B.Sc. degree in physics in 2010 from the University of Warsaw. He is employed at the Institute of Meteorology and Water Management - National Research Institute. His current work is focused on coupling of the numerical model EULAG with the German/Swiss weather prediction model COSMO.



Roman Wyrzykowski received M.Sc. and Ph.D degrees from the Kiev Polytechnic Institute in Computer Science in 1982 and 1986, respectively. Since 1982, he is employed at the Czestochowa University of Technology, Poland, where currently he is the head of Department of Computer and Information Science, and Director of Metropolitan Area Network in Czestochowa. His fields of expertise are parallel and distributed computing, mapping algorithms onto

parallel architectures, cluster and cloud technologies with applications. Prof. Wyrzykowski is the coordinator of the project “Methods and algorithms for organization of computations in the class of anelastic numerical models for geophysical flows on modern computer architectures with realization in the EULAG model” funded by the National Science Centre.



Krzysztof Rojek received his M.Sc. in computer SCIENCE from the Czestochowa University of Technology in 2008 and his PhD in 2012. During this period, his doctoral research focused on adaptation of high performance computing to the parallel processors architectures including Cell Broadband Engine and GPUs. Since 2012, Dr. Rojek is employed at Czestochowa University of Technology. His current work is focused on automated performance tuning and

adaptation of the EULAG numerical model to the graphics processor architectures.