Section 3

Computational studies including new techniques, the effect of varying model resolution, parallel processing

Numerical experiments with finite difference schemes using data of regional atmospheric model COSMO-RU

D. Yu. Alferov (dalferov@mecom.ru), G. S. Rivin (Gdaly.Rivin@mecom.ru)
Hydrometcenter of Russia, Moscow, Russia

There are three basic classes of finite-difference schemes used for numerical weather prediction: explicit, implicit, and semi-implicit. A large series of numerical experiments demonstrate some advantages of semi-implicit methods under certain conditions. Some work has been already made to introduce semi-implicit methods into the regional atmospheric model COSMO-RU [3]. In order to find out the effect of this modification we have compared the explicit and semi-implicit schemes for a barotropic model using the COSMO-RU objective analysis data and boundary conditions updating with interpolated hourly results of 24 hour COSMO-RU prediction on each step. This article contains a brief description and results of this work.

Equations and numerical schemes

The barotropic model is represented by the following equations in the Cartesian coordinate system:

$$\frac{\partial u}{\partial t} - Qv + \frac{\partial K}{\partial x} = -\frac{\partial \Phi}{\partial x},$$

$$\frac{\partial v}{\partial t} + Qu + \frac{\partial K}{\partial y} = -\frac{\partial \Phi}{\partial y},$$

$$\frac{\partial \Phi}{\partial t} + \frac{\partial (\Phi u)}{\partial x} + \frac{\partial (\Phi v)}{\partial y} = 0,$$

$$K = \frac{1}{2}\left(u^2 + v^2\right),$$

$$Q = f + \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}.$$

Here u and v are wind components in the x- and y-directions, Φ is the geopotential, f is the Coriolis parameter, K is the kinetic energy, and Q is the vertical component of the absolute vorticity.

The Arakawa A grid was used for the explicit scheme, while the Arakawa B grid was applied for the semi-implicit scheme.

Filtering methods

To avoid computational noise in data fields near the boundaries, we can move several grid rows closer to the boundaries. For better noise suppression, we can use weighted averaging of neighboring values. We use the following formula:

$$\varphi^*_{k,l} = \frac{a\varphi_{k,l} + b\left(\varphi_{k,l+1} + \varphi_{k-1,l+1} + \varphi_{k+1,l} + \varphi_{k+1,l+1} + \varphi^*_{k,l-1} + \varphi^*_{k-1,l-1} + \varphi^*_{k-1,l} + \varphi^*_{k+1,l-1}\right)}{100},$$

$$a < 100,\ b = \frac{100 - a}{8},$$

where φ is the initial field and φ^* is the modified field.

COSMO-RU model

The COSMO-RU model [1, 2, 4] uses a second-order leapfrog HE-VI (horizontally explicit, vertically implicit) and a two time-level second- and third-order Runge-Kutta split-explicit schemes. A three time-level 3-dimensional semi-implicit scheme is also inserted into the model, but has not been properly tested yet. The variant of the model used for comparison in this work uses a 350×310 grid with a grid step of 14 km in rotated spherical coordinates (rotated coordinates of the southwest corner are 19° S, 19° W, and the geographical coordinates of the rotated North pole are 35° N, 145° W).

Numerical experiments

We took the 500 hPa geopotential height and horizontal wind components (for 2010/01/30) at 500 hPa as initial data.

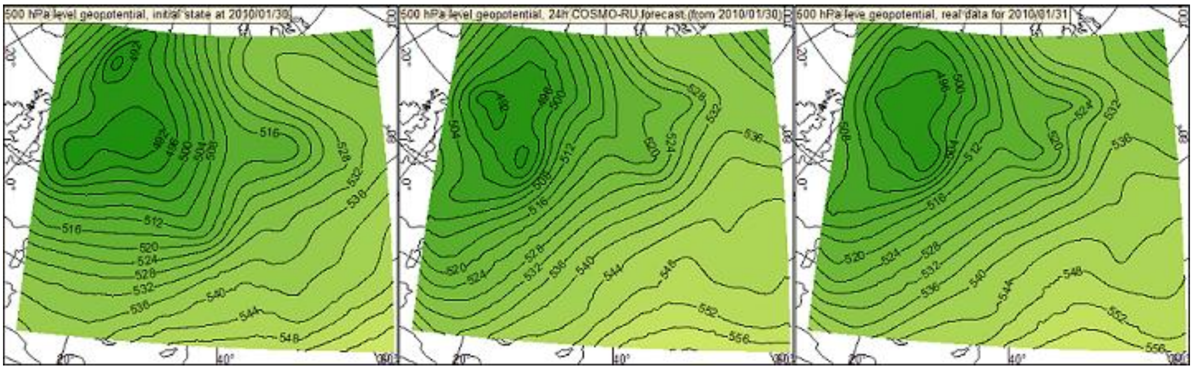
500 hPa level geopotential, initial state at 2010/01/30
500 hPa level geopotential, 24h COSMO-RU forecast (from 2010/01/30)
500 hPa leve geopotential, real data for 2010/01/31

Fig. 1. 500 hPa geopotential height (dam): initial state at 00 UTC 2010/01/30, 24-h COSMO-RU forecast (using a second-order leapfrog HE-VI scheme) and COSMO-RU objective analysis at 00 UTC 2010/01/31. The model time step is 80 seconds.

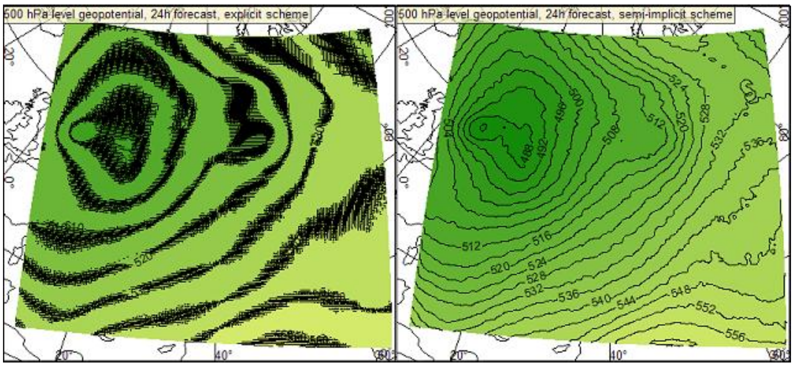
500 hPa level geopotential, 24h forecast, explicit scheme
500 hPa level geopotential, 24h forecast, semi-implicit scheme

Fig. 2. Left picture: The 24-hour forecast of 500 hPa geopotential height (dam) using the explicit scheme. The maximum difference from the corresponding objective analysis is about 17 dam. We used here the averaging filter at every 7th step. The time step was 18 sec. Right picture: The same but for the semi-implicit scheme. The maximum difference from the objective analysis is about 15.6 dam. The time step was 180 sec, and no filtering was applied.

As we see from these experiments, the semi-implicit scheme shows the best results even with much larger step (180 seconds instead of 18). So we can say that the semi-implicit scheme can be useful to make the calculations in the COSMO-RU model faster. In the nearest future we plan to introduce the semi-implicit method into the COSMO-RU model.

References

1. Doms G., Schaettler U. *A description of the non-hydrostatic regional model LM. Part I: Dynamics and Numerics.* Deutscher Wetterdienst, November 2002, 134 p.
2. Schaettler U., Doms G., Schraff C. *A description of the non-hydrostatic regional model LM. Part VII: User's Guide.* Deutscher Wetterdienst, December 2009, 142 p.
3. Vilfand R. M., Rivin G. S., and Rozinkina I. A. *Mesoscale short-term regional weather forecasting in the Hydrometeorological center of Russia on the example of COSMO-RU model.* Russian Meteorology and Hydrology, 2010, Vol. 35, No. 1.
4. http://www.cosmo-model.org/

A big brother multi-pole experiment with stretched grid GCMs

Michel Déqué
Centre National de Recherches Météorologiques (CNRS/GAME), Météo-France.
42 avenue Coriolis F-31057 Toulouse Cédex 1, France, deque@meteo.fr

Limited domain and global variable resolution models are the two cheap techniques to produce a full dynamical climate simulation at high resolution. Each technique has its advantages and drawbacks. Although limited domain models (e.g. Giorgi 1990) are always based on a rectangular domain surrounded by thin rectangular relaxation areas, there exist several techniques to produce variable resolution over the globe (Fox-Rabinowitz et al., 2008). As demonstrated by Courtier and Geleyn (1988), the only method which ensures isotropy everywhere is the one based on homothetic expansion on a stereographic projection plane. Isotropy of the horizontal discretization is a useful property because the atmosphere equations (except the Coriolis term) are independent of the direction of the axes. So, introducing anisotropy in the numerics introduces artifacts in the solutions, e.g. by filtering more waves coming from the West than waves coming from the North. On the other hand this technique is little flexible with regard to the domain definition. Indeed, as soon as the pole of high resolution and the total number of points are defined, the only degree of freedom is the stretching factor: with a high stretching factor, you get high resolution near the pole, but the resolution decreases rapidly when going to the antipodes. It is not possible to combine high resolution over a wide area and low resolution over the rest of the globe, with a reasonable transition in between.

In Déqué (2010) two experiments are presented in a "perfect model" framework. A high resolution global GCM is supposed to be a reference that several stretched-grid GCMs try to mimic. In the first experiment, the GCMs are run without constraint. One observes that the statistical characteristics are tiling rather nicely: the grid points can be pooled so that the different stretched runs produce a quasi-seamless distribution over the globe. However, when a single day is considered, the "GCM-mosaic" is no longer seamless, because each stretched simulation is independent. In a second experiment, the stretched GCMs are driven by a low resolution GCM in the part of their grid where resolution is low. Then, as can be seen for example by Hovmoeller diagrams, the "GCM-mosaic" is almost seamless on a day-to-day basis. The added value of the "mozaic GCM" with respect to the low resolution driving GCM can be evaluated by comparing to the high resolution GCM. However, this comparison can be done only in terms of statistical distribution, because the chronology of the high and low resolution runs are independent.

In order to evaluate the ability of the re-created high resolution to mimic the reference, the big brother approach (Denis et al., 2002) is a traditional approach in regional climate modeling. This is the aim of the present study. Here the big brother is a TL511 (40 km mesh) version of ARPEGE-Climate version 5.1 (Déqué 2010). A 20-year simulation with monthly observed SSTs (1989-2008) is performed, and the 6-hourly prognostic variables are filtered to TL127. This big brother has five little brothers in TL127 (160 km mesh) with a stretching factor of 4, so that the maximum resolution of each little brother is equal to the resolution of the big brother. The five poles are located in the Atlantic-Europe domain, so that the resolution is almost constant: (69.5W,35.5N), (45W,35.3N), (26.2W,49.7N), (0W,40.8N), (26.2E,49.7N).

Figure 1 shows the resolution of the mosaic based on the 5 stretched grids. The dots correspond to the grid points where a given little brother is not relaxed towards the big brother. Thus the dotted area is the equivalent free area of a limited area model. The relaxation time depends on resolution: it decreases from 1 day (resolution 70 km) to 1 time step where the resolution of the little brother is equal to the resolution of the driving conditions (160 km). The driving is much more progressive than in a limited area model, reducing the risk of numerical shock at the boundaries.

Figure 2 shows the DJF mean and standard deviation of mean sea level pressure of the high resolution GCM and the 5 stretched GCMs driven by the filtered output of the high resolution GCM. They are really similar and the seams (guessed from Fig. 1) are hardly visible. The daily correlation between the two models is 0.99. If we apply to the daily fields the same filtering as for the driving conditions to extract the high resolution part, we still have an anomaly correlation of 0.26 in the free zone (dotted area of Fig. 1) between little and big brothers. This shows that the small scales regenerated by the little brothers contain a part of information common with the big brother's small scales.

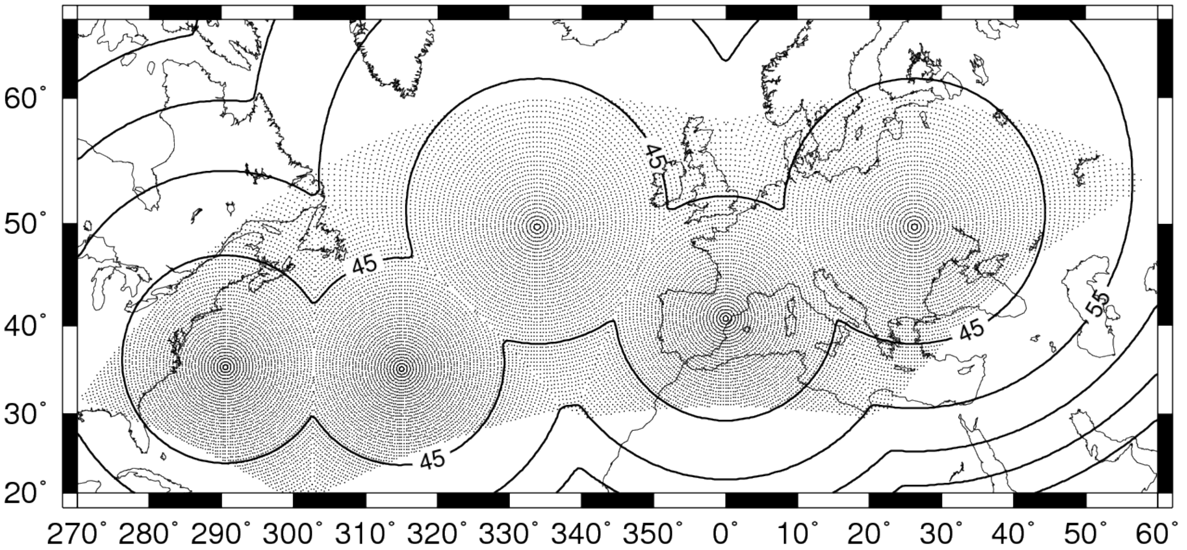

Figure 1: Equivalent resolution of the mosaic obtained with the 5 little brothers (contour interval 10 km); the dots correspond to individual grid points which are not relaxed in the stretched GCMs.

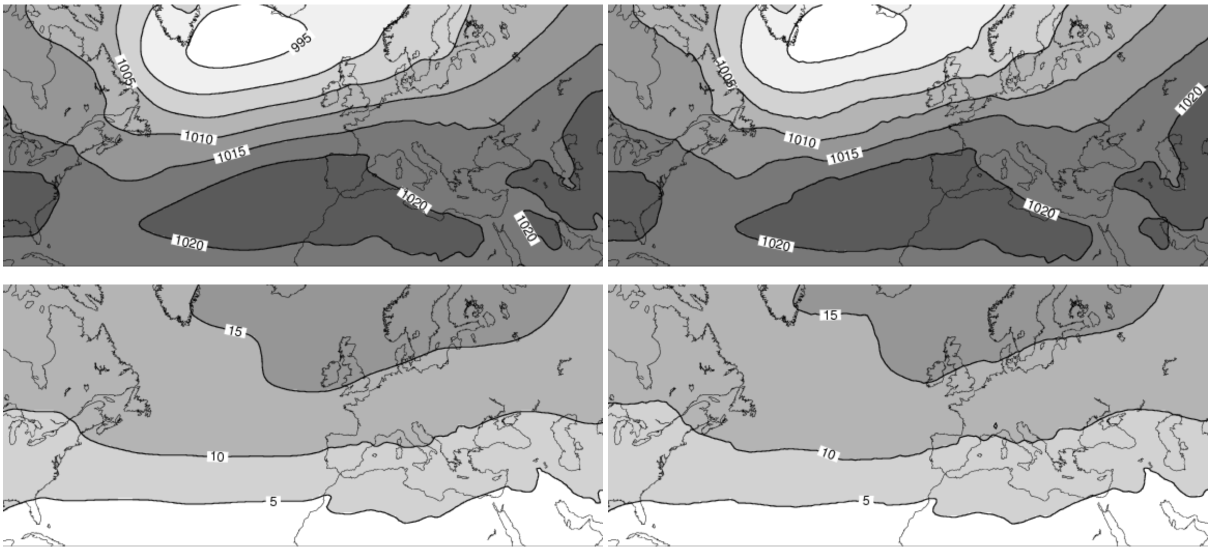

Figure 2: DJF mean sea level pressure of big brother (left) and mosaic of little brothers (right); mean (top) and standard deviation (bottom); contour interval 5 hPa.

References:

Courtier P. and J.F. Geleyn, 1988: A global numerical weather prediction model with variable resolution: Application to the shallow water equations. *Quart. J. Roy. Meteor. Soc*. **114**, 1321-1346.

Denis B., R. Laprise, D. Caya and J. Côté, 2002: Downscaling ability of one-way nested regional climate models: the Big-Brother Experiment. *Clim. Dyn.* **18**, 627-646.

Déqué M., 2010: Regional climate simulation with a mosaic of RCMs. *Meterol. Zeitschrift*, accepted.

Fox-Rabinowitz M., J. Côté, B. Dugas, M. Déqué, J.L. Mc Gregor and A. Belochitski, 2008: Stretched-grid Model Intercomparison Project: decadal regional climate simulations with enhanced variable and uniform-resolution GCMs. *Meteorol. Atmos. Phys*. **100**, 159-177.

Giorgi F., 1990: Simulation of regional climate using a limited area model nested in a general circulation model. *J. Climate* **3**, 941-963.

Semi-Lagrangian mass-conservative advection scheme on the sphere on the reduced grid

Vladimir Shashkin
Moscow Institute of Physics and Technology 9 Institutski per. 141700 Dolgoprudniy Russia
E-mail: vvshashkin@yandex.ru

Mikhail Tolstykh
Hydrometcentre of Russia
and
Institute of Numerical Mathematics RAS, 8 Gubkina st. 119333 Moscow Russia
E-mail: tolstykh@inm.ras.ru

1. Introduction

Current climate models require efficient schemes for advection of humidity, liquid and solid water variables, and a number of chemical constituents. Semi-Lagrangian (SL) transport schemes have proved to be an efficient numerical method for treating the advection process. However, a serious disadvantage of most SL schemes is that they do not formally conserve integral invariants, in particular, total mass, which has been found to drift significantly if no corrections are applied during long integrations of the SL climate model. Finite-volume based conservative SL transport schemes have gained prominence during the recent years, but only a few of them are available for spherical geometry (global) application. The most successfull developments in the spherical geometry are presented in papers [1], [2], [3]. Current work generalizes the local conservative cascade semi-Lagrangian global advection scheme introduced in [2] to the case of latitude-longitude reduced grid (see sect. 3 for details).

2. The finite-volume based conservative SL advection schemes

Unlike traditional SL schemes, finite-volume based conservative SL schemes use grid cells rather than grid points and cell-averaged values of density rather than its grid-point values. Backward trajectories with arrival points at regular (Eulerian) grid cell corners are constructed to define departure (Lagrangian) cell. The cell averaged density value in the j-th grid cell on the time level $n+1$ is defined as follows: $\bar{\rho}_j^{n+1} = \frac{M^n(A_j^*)}{mes(A_j)}$. Here $M^n(A_j^*)$ is the mass enclosed on the time level n in A_j^* - the departure (Lagrangian) cell corresponding to the arrival cell coinciding with j-th grid cell and $mes(A_j)$ is the square of j-th grid cell. Thus, the keypoint of finite-volume based conservative SL schemes is to calculate mass enclosed in each Lagrangian cell.

3. Conservative cascade scheme on the reduced grid

Latitude-longitude reduced grid is quite similar to the regular latitude-longitude grid, the difference is that the number of points in latitude rows decreases toward the poles. So, the latitude resolution is uniform and the longitude resolution decreases towards the poles remaining uniform inside each latitude row.

In order to apply the conservative cascade scheme on the reduced grid firstly the density is redistributed from the reduced grid to the regular (full) latitude-longitude grid in conservative manner. The latitude rows of the full grid coincide with those of reduced grid, and the number of points in latitude rows of the full grid is equal to the number of grid-points in equatorial latitude row of the reduced grid. This means that series of conservative 1D remappings (the remapping technique is described in [4]) inside latitude rows should be done to obtain cell-averaged values of density on the full grid. These values are then used to estimate the masses enclosed in the Lagrangian cells via conservative cascade scheme for the regular grid. The only dissimilarity is that the number of Lagrangian cells to be treated in the 2nd remapping differs from row to row. The meridional Courant number should be less than .5 in polar regions for the scheme to works correctly (see [2] for details). However, modification obviating this restriction is going ahead.

4. Numerical Experiments

a. Solid body rotation. The scheme was tested on the solid body rotation problem (test #1 from [6]). The initial distribution was the cosine-bell. Numerical experiments were carried out on the regular grid with resolution of 1.5^0 and on reduced grid from [5] of the same maximum resolution, which have 10% less points than regular grid. The angle of solid rotation $\alpha = \frac{\pi}{2}$, the center of the bell was chosen such that distribution goes along the pole to pole direction. Full revolution required 480 steps (meridional Courant number $C_\theta = 0.5$). All other parameters were set up as in [2]. The exact solution after one revolution is just the initial distribution. Exact backward trajectories were used. The results are presented in Table 1 and on Figure 1.

b. Smooth deformational flow. The scheme was tested on the smooth deformational flow problem (see [2] for full description). All parameters except resolution and time step were chosen as in [2]. The grids were the same as in the solid body rotation test. The test problem was integrated for 3 time units (nondimensional) with 120 time steps. The results are presented in Table 2.

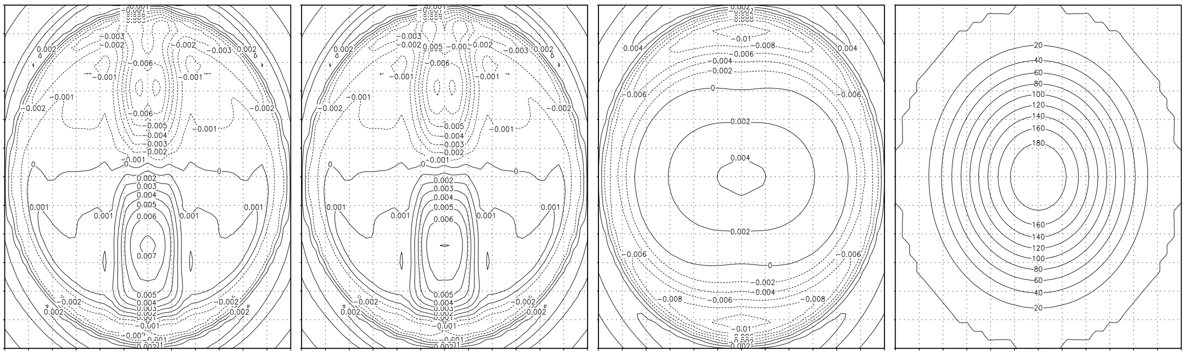

Figure 1: Error fields in solid rotation test after one revolution for conservative cascade scheme on full (a) and reduced grids (b), error field for non-conservative SL scheme (c), initial distribution (d)

Scheme	l_1	l_2	l_∞	max
Conservative cascade scheme on reduced grid	1.97E-02	1.34E-02	1,34-02	-0.4E-02
Conservative cascade scheme on regular grid	1.97E-02	1.34E-02	1,34-02	-0.4E-02
Non-conservative SL scheme	6.03E-02	3.78E-02	3,1E-02	-1.8E-02

Table 1: Error measures for solid body rotation test

Scheme	l_1	l_2	l_∞	max
Conservative cascade scheme on reduced grid	2.3E-04	6.3E-04	8.0-03	-1.45E-07
Conservative cascade scheme on regular grid	2.3E-04	6.3E-04	6.4-03	-1.45E-07

Table 2: Error measures for smooth deformational flow test

The cascade conservative SL scheme exactly preserves the mass and is also more accurate than non-conservative SL scheme.

Currently the presented scheme is being implemented in the shallow-water model on the sphere [7].

This work was supported with the Russian RFBR grant 10-05-01066.

References

1. Nair R.D., Machenhauer B. The mass-conservstive cell-integrated semi-Lagrangian advection scheme on the sphere // Mon. Wea. Rev. 2002 V. 130 P. 649-667.

2. Nair R.D. [et al.]. Efficient conservative global transport scheme for climate and atmospheric chemistry models // Mon. Wea. Rev. 2002 V. 130 P. 2059-2073.

3. Zerroukat M. [et al.]. SLICE-S: A semi-Lagrangian inherently conserving and efficient scheme for transport problems on the sphere // Quart. J. Roy. Meteor. Soc. 2004 V. 130, N 602 P. 2649-2664.

4. Colella P., Woodward P.R. The picewise parabolic method (PPM) for gas-dynamical simulations // J. Comput. Phys. 1984 V. 54 P. 174-201.

5. Fadeev R.Yu., Construction of a reduced latitude-longitude grid for a global numerical weather prediction problem. // Russian Meteorology and Hydrology, 2006, N 9, p. 5-20.

6. Williamson, D. L., Drake, J. Hack, R. Jacob, and P. N. Swartztrauber, 1992: A standard test for numerical approximation to the shallow water equations in spherical geometry. // J. Comput. Phys., 102, 211 224.

7. Tolstykh M. Vorticity-divergence semi-Lagrangian shallow-water model on the sphere based on compact finite differences // J. Comput. Phys. 2002. V. 179. P. 180-200.

GPU Acceleration of the Meso-scale Atmospheric Model "ASUCA"

Takashi Shimokawabe[1], Takayuki Aoki[1], Junichi Ishida[2], Chiashi Muroi[2]
[1] *Tokyo Institute of Technology, Japan;* [2] *Japan Meteorological Agency*
E-mail : shimokawabe@sim.gsic.titech.ac.jp

1 Introduction

Graphics Processing Units (GPUs) offer high performance of floating point calculation and wide memory bandwidth. Recently, general-purpose computation on GPUs (GPGPU) has become an active area of research in parallel computing because they provide high performance at relatively low cost for scientific computing applications. In the field of high performance computing, it was reported that various applications such as computational fluid dynamics [1] and astrophysical N-body simulations [2] ran dozens of times faster on a GPU than on a CPU core. In the field of numerical weather prediction, GPU acceleration of several modules from the Weather Research and Forecast (WRF) model were reported. Michalakes et al. reported a 20× speedup by GPU computing for WRF Single Moment 5-tracer (WSM5) microphysics, which is a computationally intensive physics module of the WRF model [3], but this speedup remains as a 1.3× overall improvement in the application. Linford et al. reported a 8.5× increase in speed on a GPU for a computationally expensive chemical kinetics kernel from WRF model with Chemistry (WRF-Chem) as compared to serial implementation [4]. Module-by-module acceleration is adopted as an approach to increase WRF speeds.

Full GPU application, in which all calculations are executed on a GPU using variables allocated on its memory, is essential in achieving more than ten times acceleration over the whole application compared to CPU application. This allows simulation to be run without frequent data transfer between the GPU and the host computer.

We are currently working on full GPU application for ASUCA [5] - a next-generation high resolution meso-scale atmospheric model being developed by the Japan Meteorological Agency. As a first step, we have implemented its dynamical core as a full GPU application, representing an important step toward establishing an appropriate framework for full GPU-based ASUCA. The GPU code is written from scratch in the CUDA (Compute Unified Device Architecture) [6] using its original code in Fortran as a reference. The Numerical results obtained from the GPU code agree with those from the CPU code within the margin of machine round-off error. In this paper, we report the results of GPU acceleration of the dynamical core in ASUCA, which has not yet been accelerated in WRF.

2 GPU implementation

In this study, we computed the dynamical core on an NVIDIA GTX 285 using CUDA 2.3 with an AMD Phenom 9750 Quad (2.4 GHz) and 8 GByte of memory as the host computer.

In the CUDA programming, CUDA kernels for the GPU are programmed. When a kernel is launched, it is executed by individual threads arranged into blocks with unique block and thread IDs. All blocks are grouped as a grid and all threads in the grid are able to access VRAM called as the global memory on the GPU. Access to the global memory takes 400 to 600 clock cycles, which corresponds to 159 GB/s, for example, in the case of the GTX 285. 16 kByte of shared memory is assigned in each block as scratchpad memory with an access time of about two cycles. Any part of the shared memory can be read and written by all threads in the block, which is utilized as a software-managed cache to reduce access to the global memory.

The implementations of some functions used in the dynamical core in ASUCA are explained here.

2.1 Advection

In order to improve calculation performance in ASUCA, access to the global memory is reduced by making use of the shared memory as a software-managed cache. To calculate advection for a given grid size (nx, ny, nz), the kernel functions for the GPU are configured for execution in $(nx/64, nz/4, 1)$ blocks with $(64, 4, 1)$ threads in each block. Each thread specifies an (x, z) point and performs calculations from $j = 0$ to $j = ny - 1$ marching in the y direction. In order to facilitate the implementation of kernel functions for domain decomposition with MPI, the z direction in numerical space is mapped to the y direction in CUDA.

The four-point stencil of a point in each direction is required to compute advection. To carry out calculations in the j th slice, the elements in the current slice are needed for calculations by more than one thread. On the other hand, elements preceding and succeeding the current y position are used only for the thread corresponding to the element's (x, z) position. Each block therefore has an array with $(64 + 3) \times (4 + 3)$ elements in its shared memory, which is utilized to accommodate 2D sub-domain data and halos for the current j th slice. The elements along the y axis are stored in registers on the corresponding thread (Figure 1). Data stored in both shared memory and registers to perform the j th computation are reused for the $j+1$ th calculation as far as possible.

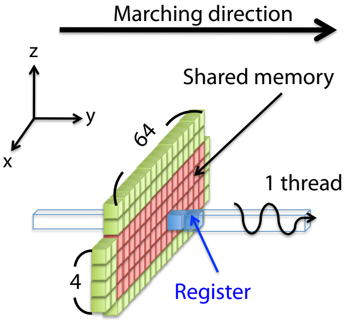

Figure 1: $(64+3) \times (4+3)$ elements in shared memory and 3 elements in registers along the y axis

2.2 1D Helmholtz-like elliptic equation

The 1D Helmholtz-like elliptic equation is solved in the vertical direction because the HE-VI scheme is adopted in ASUCA. Through discretization of the equation, a tridiagonal matrix is obtained. The basic strategy of implementation for the solver for this matrix is the same as that for advection. However, because sequential computation in the z direction is required for it, threads should not march along the y axis in view of the efficiency of parallel computing by threads. Thus, $(nx/64, ny/4, 1)$ blocks with $(64, 4, 1)$ threads are configured to the given grid size (nx, ny, nz). The threads march in the z direction for the Helmholtz-like elliptic equation.

3 Results

Figure 2 shows the performance of the dynamical core in ASUCA in both single- and double-precision floating-point calculation for six different grid sizes. With nx set as 32 and nz set as 64, the value of ny is varied from 16 to 56. In order to measure the performance on a GPU, we count the number of floating-point operations in ASUCA running on a CPU with the Performance API (PAPI) [7]. This code is implemented in C/C++ language corresponding to GPU code. Using the obtained count and the GPU computation time, the performance on the GPU is evaluated. The performance of 67.1 GFlops in single precision for a $320 \times 56 \times 64$ mesh on a single GPU has been achieved. It is found that the dynamical core in ASUCA implemented on the GPU runs 51.5 times faster than the original code for CPU performed by the serial implementation in Fortran on the Intel Core i7 920 2.67 GHz. In the case of the computation in double precision, the speed is increased by a factor of 15.8.

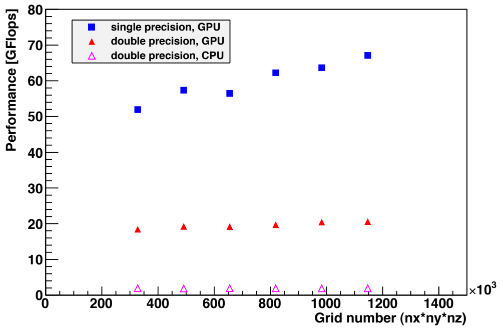

Figure 2: Performance of ASUCA on a GPU (GTX 285) and a CPU. The solid blue and red points indicate the performance of the GPU version in single and double precision respectively. The magenta outline points show the performance of the original Fortran code running on a CPU core.

4 Conclusion and future work

We are currently developing a full GPU version of ASUCA. As a first and key step, we have implemented its dynamical core on a GPU. The effective utilization of shared memory in the GPU for optimization has resulted in the performance of 67.1 GFlops, which is 51.5 times faster than the original code on a CPU. Implementation for multi-GPUs will be a subject of future work.

Acknowledgments This research was supported in part by the Global Center of Excellence Program "Computationism as a Foundation for the Sciences" from the Ministry of Education, Culture, Sports, Science and Technology of Japan and in part by the Japan Science and Technology Agency CREST research program "Ultra Low-Power (ULP-HPC)".

References

[1] J. C. Thibault and I. Senocak. CUDA implementation of a navier-stokes solver on multi-GPU desktop platforms for incompressible flows. In *Proceedings of the 47th AIAA Aerospace Sciences Meeting*, number AIAA 2009-758, jan 2009.

[2] Tsuyoshi Hamada, Tetsu Narumi, Rio Yokota, Kenji Yasuoka, Keigo Nitadori, and Makoto Taiji. 42 tflops hierarchical n-body simulations on gpus with applications in both astrophysics and turbulence. In *SC '09: Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, pages 1–12, New York, NY, USA, 2009. ACM.

[3] John Michalakes and Manish Vachharajani. GPU acceleration of numerical weather prediction. In *IPDPS*, pages 1–7. IEEE, 2008.

[4] John C. Linford, John Michalakes, Manish Vachharajani, and Adrian Sandu. Multi-core acceleration of chemical kinetics for simulation and prediction. In *SC '09: Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, pages 1–11, New York, NY, USA, 2009. ACM.

[5] Junichi Ishida, Chiashi Muroi, Kohei Kawano, and Yuji Kitamura. Development of a new nonhydrostatic model "ASUCA" at JMA. *CAS/JSC WGNE Reserch Activities in Atmospheric and Oceanic Modelling*, 2010.

[6] CUDA Programming Guide 2.3. `http://developer.download.nvidia.com/compute/cuda/2_3/toolkit/docs/NVIDIA_CUDA_Programming_Guide_2.3.pdf`, 2009.

[7] S. Browne, J. Dongarra, N. Garner, G. Ho, and P. Mucci. A portable programming interface for performance evaluation on modern processors. *Int. J. High Perform. Comput. Appl.*, 14(3):189–204, 2000.