Geosci. Model Dev., 5, 345–353, 2012
www.geosci-model-dev.net/5/345/2012/
doi:10.5194/gmd-5-345-2012
© Author(s) 2012. CC Attribution 3.0 License.

Geoscientific Model Development

ASAMgpu V1.0 – a moist fully compressible atmospheric model using graphics processing units (GPUs)

S. Horn

Leibniz Institute for Tropospheric Research, Permoserstrasse 15, 04318 Leipzig, Germany

Correspondence to: S. Horn (horn@tropos.de)

Received: 12 September 2011 – Published in Geosci. Model Dev. Discuss.: 7 October 2011
Revised: 23 January 2012 – Accepted: 5 March 2012 – Published: 20 March 2012

Abstract. In this work the three dimensional compressible moist atmospheric model ASAMgpu is presented. The calculations are done using graphics processing units (GPUs). To ensure platform independence OpenGL and GLSL are used, with that the model runs on any hardware supporting fragment shaders. The MPICH2 library enables interprocess communication allowing the usage of more than one GPU through domain decomposition. Time integration is done with an explicit three step Runge-Kutta scheme with a time-splitting algorithm for the acoustic waves. The results for four test cases are shown in this paper. A rising dry heat bubble, a cold bubble induced density flow, a rising moist heat bubble in a saturated environment, and a DYCOMS-II case.

1 Introduction

During the second half of the last century large eddy simulations (LES) enabled researchers to gain insight into various questions concerning turbulent structures in convective and stable boundary layers. The term LES is widely used for models with grid sizes from centimeters to hundreds of meters that include subgrid scale turbulence parameterization. The coarse scale LES were also used to study moist processes like clouds and drizzle in the planetary boundary layer. The resolution of regional meteorological models reached a point where atmospheric convection at least in larger convective systems can be resolved. The increased resolution of regional models and the growing domain sizes for LES models require faster computational architectures. In the last years there were increasing efforts to use graphics processing units (GPUs) to speedup these kind of models. For example in the weather research forecast model (Skamarock et al., 2005) several parts of the model like the microphysical parameterization, passive tracer advection and the chemical solver were migrated to the GPU (Michalakes and Vachharajani, 2008). Problems with high data transfer costs are documented if parts of a complex model are migrated to GPUs. Those limit possible speedups. Another approach is to run the model completely on the GPU to avoid this data transfer. The model GALES (Schalkwijk et al., 2012) written using CUDA uses this approach. In this work, the model ASAMgpu is presented, which is focused at the scale between LES and Cloud Resolving Models (CRMs) and uses Graphics Processing Units (GPUs) as the computational infrastructure. The used fundamental equations are the fully compressible Euler equations, integrated forward in time using an explicit three-step Runge Kutta scheme with a time-splitting algorithm. As the prognostic thermodynamic variable describing the energy contained in the system, a quantity related to the total entropy was chosen, which is conserved during isentropic processes and where temperature and pressure can be calculated explicitly. This quantity is described in more detail in the third chapter. Warm-phase microphysical processes are parameterized using a two moment scheme from Seifert and Beheng (2005). The two moments are the mass density and the number concentration for cloud water droplets and rain drops. Ice-phase microphysics are not implemented yet and the stress tensor is neglected as well, that means currently no subgrid scale model is applied. In this work results for three common test cases are presented: a rising heat bubble from Wicker and Skamarock (1998), a falling cold bubble by Wicker and Skamarock (2001) and a moist heat bubble in a supersaturated environment driven by latent heat release after (Bryan and Fritsch, 2002). This experiment also addresses the condensation rate and the latent heat release. The last section is about one more complex test scenario for high resolution atmospheric models,

a nocturnal drizzling stratocumulus layer (Wang and Feingold, 2009) and results of the first scaling experiments using this case are presented. Especially for the last example a large domain size at high resolution is necessary and the integration time is 16 h at a time step of one second. To solve this computational demanding task in a tolerable time a powerful computer is needed. Therefore one quad core CPU node with four GPUs (AMD Radeon HD 5870) is used. The CPU is an Intel(R) Xeon(R) CPU E5530 with 4 cores, with that 4 GPU threads can be distributed over the available cores. One of these GPUs has an theoretical peak performance of 2.72 TFlops (ATI Specs, 2009), with that four of them should be about 10 TFlops, but this is a very optimistic value disregarding communication cost between GPUs. The electric power consumption of this system is in the range of one kilowatt. To provide the possibility to run a cluster of these nodes the model is also designed for inter-node communication using the second version of the Message Passing Interface (MPICH2, Gropp et al., 1999). The operating system is a standard linux with hardware accellerated xserver using the proprietary drivers by AMD. Currently it is not possible to run the model on a CPU only system.

2 OpenGL and GLSL

The model uses modern Graphics Processing Units (GPUs) for fast and efficient massive parallelized computations. The application of GPUs for non-graphic calculations is also called General Purpose Computation on Graphics Processing Units (GPGPU). Different approaches for GPGPU exist. Some of them are vendor specific like CUDA from nVidia or Stream from ATI depending on the corresponding hardware. Vendor independent solutions currently available are OpenGL with the OpenGL Shader Language (GLSL) and OpenCL. This model was developed using OpenGL + GLSL.

The main concepts of all GPGPU solutions are quite similar, but the vocabulary is different between the different approaches. The computation process is defined using a certain number of buffers, also called textures. These buffers are used to store the data describing the physical system in the graphics device memory. To describe the computation applied to these buffers, small programs called kernels or shaders are used. In the OpenGL approach these kernels are written in GLSL, a language with a C-like syntax. GLSL supports functions, if-branches and loops. Branches should be used carefully because they could significantly slow down the computation step. After the definition of the used buffers and the kernel, a calculation step is initiated. This happens by rendering a plane using up to eight input textures into a number of output texture framebuffers. The number of simultaneous possible input and output textures differ from device to device, currently eight for each are a common number. During this step the graphics device splits the output buffers in a large number of blocks which then are all processed in parallel by the shader units, also called stream processors. In OpenGL this is done by the device driver completely transparent to the developer. In OpenCL and vendor specific solutions like CUDA it is possible to control block sizes and the number of used threads. Current GPUs have a large number of shader units, e.g. an ATI Radeon HD 5780 has 320 independent stream processors where every single one is a small 5-D vector unit, or e.g. nVidia Tesla C2070 with 448 scalar stream processors.

3 Used entropy variable

To describe atmospheric processes including heat fluxes, radiation and phase transitions, the equations of momentum and mass conservation have to be extended by a prognostic equation describing the transport and the sources and sinks for energy. This can be done using different thermodynamic variables, like the total energy, temperature, potential temperature, equivalent potential temperature or entropy. During this work a variable was chosen, which has no sources or sinks during isentropic processes, like advection, and from which the absolute temperature and the pressure, needed for boundary conditions and microphysics, may be derived explicitly.

The first law of thermodynamics, written in specific quantities, including phase transition from water vapor to liquid water yields

$$du(s,\alpha,\rho_l,\rho_v) = Tds - pd\alpha + \frac{1}{\rho}(\mu_v - \mu_l)d\rho_v \tag{1}$$

with the definitions of enthalpy and latent heat

$$h = u + p\alpha \quad \text{and} \quad L_v = (\mu_v - \mu_l) \tag{2}$$

we get

$$dh = Tds + \alpha dp + \frac{1}{\rho}L_v d\rho_v \tag{3}$$

Introducing the definitions of specific heat capacity at constant pressure and the gas constant for a mixture,

$$dh = c_{\text{pml}}dT \tag{4}$$

$$c_{\text{pml}} = \frac{\rho_d c_{pd} + \rho_v c_{pv} + \rho_l c_{pl}}{\rho} \quad \text{and} \quad R_{\text{ml}} = \frac{\rho_d R_d + \rho_v R_v}{\rho} \tag{5}$$

and the equation of state for a vapor air mixture

$$\alpha = \frac{R_{\text{ml}}T}{p} \tag{6}$$

we get

$$c_{\text{pml}}dT = Tds + \frac{R_{\text{ml}}T}{p}dp + \frac{L_v}{\rho}d\rho_l \tag{7}$$

or

$$ds = \frac{c_{\text{pml}}}{T}dT - \frac{R_{\text{ml}}}{p}dp - \frac{L_v}{\rho T}d\rho_l. \tag{8}$$

Integration leads to

$$s = c_{\mathrm{pml}} \ln(T) - R_{\mathrm{ml}} \ln(p) - \int \frac{L_v}{\rho T} d\rho_l + C_0 \tag{9}$$

Introducing a quantity σ as a measure for entropy content caused by temperature and pressure

$$\sigma = c_{\mathrm{pml}} \ln(T) - R_{\mathrm{ml}} \ln(p) \tag{10}$$

leas to

$$s = \sigma - \int \frac{L_v}{\rho T} d\rho_l + s_0 \tag{11}$$

For the phase transition from vapor to liquid water the assumption of conservation of total mass ($\frac{d\rho_d}{dt} = 0; \frac{d\rho_v}{dt} = -\frac{d\rho_l}{dt}$) yields

$$\frac{ds}{dt} = \frac{d\sigma}{dt} - \frac{1}{\rho}\left(c_{pl} - c_{pv}\right)\ln(T)\frac{d\rho_l}{dt} - \frac{1}{\rho} R_v \ln(p)\frac{d\rho_l}{dt} - \frac{L_v}{\rho T}\frac{d\rho_l}{dt} \tag{12}$$

with

$$\frac{ds}{dt} = \frac{1}{T}\delta Q \tag{13}$$

we get the evolution equation for σ

$$\left(\frac{\partial \sigma}{\partial t} + (\boldsymbol{v}\nabla)\sigma\right) = \frac{1}{T}\delta Q + \frac{1}{\rho}\left(c_{pl} - c_{pv}\right)\ln(T)\frac{d\rho_l}{dt} + \frac{1}{\rho} R_v \ln(p)\frac{d\rho_l}{dt} + L_v \frac{d\rho_l}{dt} \tag{14}$$

This quantity has some advantages. First it is conserved under adiabatic processes without phase transitions. Second, explicit equations for absolute pressure and temperature can be obtained. And last but not least the source terms in the case of phase transitions can be derived quite easily.

$$\theta = \exp\left(\frac{\rho\sigma}{c_{\mathrm{pml}}} + \frac{R_{\mathrm{ml}}}{c_{\mathrm{pml}}} \ln(p_0)\right) \tag{15}$$

$$T = \exp\left(\frac{\rho\sigma}{c_{\mathrm{pml}} - R_{\mathrm{ml}}} + \frac{\ln(R_{\mathrm{ml}})}{c_{\mathrm{pml}}/R_{\mathrm{ml}} - 1}\right) \tag{16}$$

$$p = \exp\left(\frac{\rho\sigma}{c_{\mathrm{pml}} - R_{\mathrm{ml}}} + \frac{\ln(R_{\mathrm{ml}})}{1 - R_{\mathrm{ml}}/c_{\mathrm{pml}}}\right) \tag{17}$$

4 Model description

4.1 Governing equations

The equations used in the GPU-Model are a form of the Euler equations for a compressible fluid in conservative form where the conservation of mass is applied for the bulk density (ρ). In addition further transport equations for the partial phases water vapor density (ρ_v), cloud water density (ρ_c) and rain water density (ρ_r), including the source terms from the microphysics ($S_{\rho_i \mathrm{MP}}$) are used. The momentum equation is the standard Euler equation using bulk density and bulk momentum. The energy equation is written in the form of entropy derived in the last chapter. In addition to the mass density transport equations, similar equations are included for available cloud condensation nuclei density (N_{CCN}), cloud droplet density (N_c) and rain droplet density (N_r), again with sources from the microphysical parameterization ($S_{N_i \mathrm{MP}}$). These source terms currently include mass transfer between the different phases and changes in number concentration density due to the processes of condensation and evaporation, activation, selfcollection, autoconversion and sedimentation.

Subgrid scale turbulence, the Coriolis force and ice-phase microphysics are currently ignored. So the basic equations can be written as follows:

$$\frac{\partial \rho}{\partial t} + \nabla(\mathbf{v}\rho) = 0 \tag{18}$$

$$\frac{\partial \rho_i}{\partial t} + \nabla(\mathbf{v}\rho_i) = S_{\rho_i \mathrm{MP}} \qquad i = v, c, r \tag{19}$$

$$\frac{\partial N_i}{\partial t} + \nabla(\mathbf{v}N_i) = S_{N_i \mathrm{MP}} \qquad i = \mathrm{CCN}, c, r \tag{20}$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla(\rho \mathbf{v} \cdot \mathbf{v}) = -\nabla p - \rho \mathbf{g} \tag{21}$$

$$\frac{\partial \rho\sigma}{\partial t} + \nabla(\mathbf{v}\rho\sigma) = S_{\rho\sigma \mathrm{MP}} \tag{22}$$

4.2 Time integration and grid structure

The time integration scheme is based on Wicker and Skamarock (2001), using an explicit three step Runge Kutta scheme (RK3) with a time splitting algorithm for the fast pressure waves, which are integrated using a simple leapfrog algorithm. The Butcher tableau of the used low-storage RK3 is shown in Fig. 1. This RK3-Leapfrog combination has to be stabilized using divergence damping.

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{3} & \frac{1}{3} & & \\ \frac{1}{2} & 0 & \frac{1}{2} & \\ \hline & 0 & 0 & 1 \end{array}$$

Fig. 1. Butcher tableau of the used Runge Kutta scheme.

The model uses a staggered grid (Arakawa-C), where the scalars are cell centered, and the velocity components are stored at corresponding faces. For the advection scheme the scalars are interpolated to cell faces using a third order upwind scheme without limiters once every Runge Kutta intermediate step. The bottle neck for GPGPU is the PCIe bus which is very slow compared to internal GPU communication, such that a small stencil strongly increases performance on multi GPU architectures. Similar time integration methods are commonly used by Bryan and Fritsch (2002), in the weather research forecast model (WRF) (Skamarock et al., 2005), and the COSMO by the german weather service (DWD).

4.3 Model structure

The model is written in C++ using object oriented programming approaches. The GPU parallelization is encapsulated in two classes, the *texture* and the *shader* class. The *texture* class is used to allocate the memory on the GPU, transfer data between CPU and GPU. It handles the loading and saving of texture fields and generates a framebuffer object to enable the texture as a render target for writing into the texture from a shader. The *shader* class loads a shader source file, sends it to the device driver for compilation and linking to generate a shader program object, which then can be used to perform calculations. The shader source files are simple ASCII text files. In addition it contains an array of pointers to texture objects to specify input and output textures for the calculation step. The *shader* class takes care of binding the needed textures to texture units, transfering texture handles to the shader program and performing a calculation step.

The node level parallelization is done by a class called *transporter* which itself uses the MPICH2 library for internode communication. The class provides two methods, one to load the data for a specified part of the model domain from the GPU memory into CPU memory and to send it to another process/node, and another method to receive this data and upload it to the GPU memory. The access to the GPU memory and the MPICH2-send and -receive procedures are implemented as asynchronous data transfers. Thus a process can download data from a GPU to the CPU while the CPU already sends data to another process, or the receiving process may upload data to the GPU while receiving another dataset from a different process in parallel.

4.4 Microphysics

The model includes a two moment microphysics, based on the work of (Seifert and Beheng, 2005) (SB2005), but not all processes described by SB2005 are included yet. The processes currently implemented are activation of cloud condensation nuclei to cloud droplets, condensation and evaporation of water vapor to/from these droplets, selfcollection of cloud droplets, autoconversion from cloud droplets to rain drops, selfcollection of rain drops, accretion of cloud droplets by rain, sedimentation and evaporation of rain drops. Collisional breakup and ice-phase microphysics as very important processes in deep convective clouds are neglected, because the scope currently lays on shallow cumulus convection. The advection without limiters and numerical errors during the condensation/evaporation process causes unphysical negative values in the prognostic mass and number density variables, so all negative densities have to be clamped to zero for the parameterizations after SB2005. All transition rates are processed by an additional limiter which ensures that negative values will be drawn back to zero and transition rates will not exceed available quantities. For this limiter the unclamped values have to be used.

4.4.1 Limiter example: activation

Detailed description for the microphysical source terms can be found in SB2005. In the ASAMgpu model these source terms are modified to ensure stability and handle negative values caused by the third order advection scheme without flux limiter. In this section this algorithm is presented using the example of activation from cloud condensation nucleis to cloud droplets. After SB2005 the activation rates are nonzero if the cell is supersaturated ($S > 0.0$), the vertical velocity is positive ($w > 0.0\,\mathrm{m\,s^{-1}}$), the gradient of supersaturation in vertical direction is positive ($dS/dz > 0.0\,\mathrm{m^{-1}}$) and the temperature is above 233.15 K ($T > 233.15\,\mathrm{K}$). In this context S is the supersaturation in percent. In SB2005 a power law is used as an empirical activation spectra to compute available CCN number density while in the ASAMgpu model the number density of available cloud condensation nuclei is a prognostic variable and the supersaturation dependency is neglected. With that the activation rate is given by

$$\frac{\partial N_{\mathrm{cSB}}}{\partial t} = N_{\mathrm{CCN}} k_{\mathrm{CCN}} \frac{1}{S} \frac{dS}{dz} w \tag{23}$$

In the ASAMgpu model this activation rate gets limited by the unclamped available cloud condensation nuclei density (N_{CCN}) using a form of the triangle inequality (Eq. 26). The result of this limiting procedure is near the (for)cing if the (lim)iter is much larger then the forcing. This is the case if a huge quantity of cloud condensation nuclei is available. If the forcing is near the limiter, the process is damped to not consume more then the available N_{CCN}. But if the limiter is negative we always get a result that compensates the negative values and draws them back to zero, while conserving mass and number density budgets.

$$\mathrm{for} = \frac{\partial N_{\mathrm{cSB}}}{\partial t} \tag{24}$$

$$\mathrm{lim} = N_{\mathrm{CCN}} \tag{25}$$

$$\frac{\partial N_{\mathrm{CCN}}}{\partial t} = \mathrm{for} + \mathrm{lim} - \sqrt{\mathrm{for}^2 + \mathrm{lim}^2} \tag{26}$$

Assuming all activated droplets contain the smallest possible drop mass of 10^{-12} kg we can calculate the mass change from water vapor to cloud water. With the condensed water mass we can determine the source from latent heat and the change in $\rho\sigma$ caused by the change of mass fractions in the gas constant and heat capacity of the mixture.

$$\frac{\partial \rho_{\mathrm{c}}}{\partial t} = 10^{-12}\,\mathrm{kg} \frac{\partial N_c}{\partial t} \tag{27}$$

$$\frac{\partial \rho\sigma}{\partial t} = L_v \frac{1}{T} \frac{\partial \rho_{\mathrm{c}}}{\partial t} + \ln(T)\left(C_{pl} - C_{pv}\right) \frac{\partial \rho_{\mathrm{c}}}{\partial t} \tag{28}$$

$$+ \ln(p)\, R_v \frac{\partial \rho_{\mathrm{c}}}{\partial t} \tag{29}$$

4.4.2 Condensation and evaporation of cloud water

The saturation adjustment technique is a common method, to calculate the amount of water vapor condensed during a timestep. Therefor all microphysical parameterizations are processed and after that the complete fraction of water vapor above the saturation level is handled as condensate. In this approach supersaturation is reduced immediately and the time integration for the microphysics has to be processed separately. In the ASAMgpu model the saturation adjustment technique was replaced by a relaxation process from vapor pressure to saturation vapor pressure. In this combination of parameterizations, condensation and activation are two competing processes, both computed parallel, both consuming water vapor. The forcing for this process is the difference between actual water vapor density and the density at saturation. The process is limited by the available cloud water, so condensation occurs at supersaturation and evaporation occurs if the gridcell is unsaturated and cloud water is available. The time scale of this process is controlled by a constant C_{cond}. If this constant is very high, we are near the saturation adjustment technique and supersaturation will be decreased nearly instantly. With a very low constant this process gets too slow and convection may be suppressed, in this case the moist bubble example presented later will produce wrong results. A constant set to one seems to be a good choice.

$$\mathrm{for} = \rho_v - (p_{vs} T / R_v) \tag{30}$$

$$\mathrm{lim} = \rho_{\mathrm{c}} \tag{31}$$

$$\frac{\partial \rho_{\mathrm{c}}}{\partial t} = C_{\mathrm{cond}} \left(\mathrm{for} - \mathrm{lim} + \sqrt{\mathrm{for}^2 + \mathrm{lim}^2} \right) \tag{32}$$

Again from the condensed/evaporated mass the necessary source/sink from latent heat release/consumption for the entropy variable has to be derived.

$$\frac{\partial \rho \sigma}{\partial t} = \left(L_v \frac{1}{T} + \ln(T)(C_{pl} - C_{pv}) + \ln(p) R_v \right) d\rho_{\mathrm{c}} \tag{33}$$

Two more simple equations are used to ensure that if no condensate exists droplet number density reduces to zero, or if condensate exists droplet number density is within the limits defined by the distribution parameters (see SB2005). The speed of this correction is controlled by the constant C currently set to $0.01\,\mathrm{s}^{-1}$. This process is a transition between available cloud condensation nuclei number density and cloud droplet number density, so evaporated droplets produce new possible cloud condensation nuclei. These corrections are reducing N_{c} if droplets get to small

$$\frac{\partial N_{\mathrm{c}}}{\partial t} = \min \left(0, \quad C \left(\frac{\rho_{\mathrm{c}}}{x_{\min}} - N_{\mathrm{c}} \right) \right) \tag{34}$$

and increasing N_{c} if droplets get to big

$$\frac{\partial N_{\mathrm{c}}}{\partial t} = \max \left(0, \quad C \left(\frac{\rho_{\mathrm{c}}}{x_{\max}} - N_{\mathrm{c}} \right) \right) \tag{35}$$

with

$$\frac{\partial N_{\mathrm{CCN}}}{\partial t} = -\frac{\partial N_{\mathrm{c}}}{\partial t} \tag{36}$$

5 Examples

In this section three two dimensional test cases and one more complex three dimensional test case were simulated with the ASAMgpu model. The first two cases are a rising heat bubble Wicker and Skamarock (1998) and the cold bubble induced density flow from Wicker and Skamarock (2001). The third case is a rising moist heat bubble in a saturated environment driven by the latent heat release (Bryan and Fritsch, 2002), which is a good test for microphysical parameterizations, but does not include transition to rain. The last three dimensional case is the DYCOMS-II case with complete two moment microphysics including drizzle and the formation of open cell structures (Wang and Feingold, 2009).

5.1 Dry heat bubble

The first test is a rising heat bubble under dry conditions. The domain size is 160×80 cells at a spatial resolution of 125 m with periodic horizontal boundary conditions. The initial state consist of an adiabatic atmosphere with a perturbation in the potential temperature. The amplitude of the perturbation is 2 K with a radius of 2 km. The bubble is located in the horizontal center of the domain at 2 km height described by

$$x_{\mathrm{c}} = 10\,000\mathrm{m} \quad z_{\mathrm{c}} = r_x = r_z = 2000\mathrm{m} \tag{37}$$

$$L = \sqrt{\left(\frac{x - x_{\mathrm{c}}}{r_x} \right)^2 + \left(\frac{z - z_{\mathrm{c}}}{r_z} \right)^2} \tag{38}$$

$$\theta' = 2\cos^2 \left(\frac{\pi L}{2} \right) \tag{39}$$

A uniform horizontal velocity of $20\,\mathrm{m\,s}^{-1}$ is applied, leading to a transport of the bubble through the whole domain and the boundaries. After 1000 s a complete cycle is fulfilled and the bubble reaches the center of the domain again. The time steps for this test case were chosen after (Jebens et al., 2009) with 2 s and 10 fast pressure steps and 7 s and 30 pressure steps as well. Divergence damping with a damping coefficient of $\nu = 0.025$ is used. The results for both time steps are equal and shown in Fig. 2. The bubble rises up, the top of the bubble reaches a height of 8 km and the solution keeps symmetric as expected.

5.2 Dry cold bubble

The second test case has a slightly larger domain as the first one and in contrast to the heat bubble the perbutation now consist of a cold pool. Again the initial state is a dry, adiabatic atmosphere with a uniform horizontal velocity field of

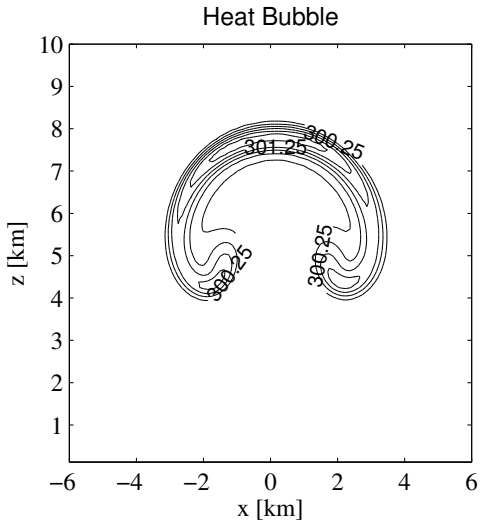

Fig. 2. Result for the heat bubble test case after 1000 s (contours: pot. temp. 0.25 K)

$20\,\mathrm{m\,s^{-1}}$. The domain now has 180×80 cells at a resolution of 200 m. With that one cylce needs 900 s. The bubble is initialized in the horizontal center of the domain again but now at 4 km height, and the horizontal radius is extended to 4 km as well. The amplitude of the pertubation is 15 K and applied to the temperature described by:

$$x_\mathrm{c} = 18\,000\,\mathrm{m} \quad z_\mathrm{c} = r_x = 4000\,\mathrm{m} \quad r_z = 2000\,\mathrm{m} \tag{40}$$

$$L = \sqrt{\left(\frac{x - x_\mathrm{c}}{r_x}\right)^2 + \left(\frac{z - z_\mathrm{c}}{r_z}\right)^2} \tag{41}$$

$$T' = -15\cos^2\left(\frac{\pi L}{2}\right) \tag{42}$$

The large timestep is 2 s and with that 6 pressure time steps are needed at this spatial scale. Again divergence damping is necessary with a damping coefficient of $\nu = 0.025$. The result after one complete cycle is shown in Fig. 3. Compared to the results of Jebens et al. (2009) and Wicker and Skamarock (2001) the overall structure is reproduced, but the solution seems to be more diffusive. That may be caused by the 3rd order upwind advection scheme, compared to the 5th order advection schemes used by the cited studies.

5.3 Moist heat bubble

The third test case for the GPU-Model is a modification of the rising heat bubble. It was suggested by Bryan and Fritsch, 2002, and is also a test for a part of the microphysics. In this case the initial state is again a hydrostatic atmosphere with neutral stability for moist air. To simplify the definition of neutral stability for the moist case two assumptions are made, the first one is the total water mixing ratio is constant and the second one is that all phase changes are exactly

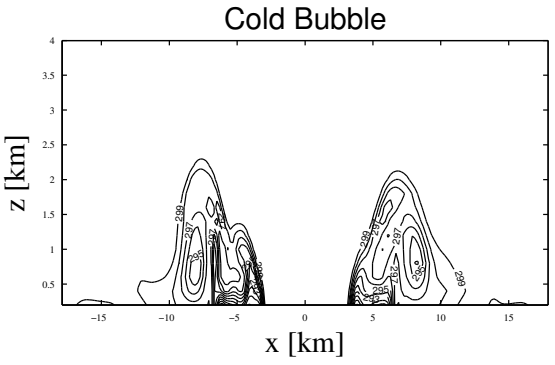

Fig. 3. Result for the cold bubble test case after 900 s (contours: pot. temp. 1 K).

reversible. Under these assumptions a moist neutral atmosphere can be defined by a constant wet equivalent potential temperature, therefore the atmosphere has to be saturated at all levels. For this test the microphysics are reduced to the reversible phase change, respectively only the processes of activation and condensation/evaporation are enabled. The simulation is a good proof of condensation rates and latent heat release, if these are too slow the bubble stops rising and will not reach the final height of 8 km after 1000 s. All other parameters are similar to the dry heat bubble test case, that includes a slightly different resolution than in the work of Bryan and Fritsch (125 m vs. 100 m), an advective timestep of 7 s with six steps for the acoustic modes and $20\,\mathrm{m\,s^{-1}}$ horizontal wind. The perturbation for a 300 K background potential temperature is applied to the density potential temperature in the form

$$x_\mathrm{c} = 10\,000\,\mathrm{m} \quad z_\mathrm{c} = 2000\,\mathrm{m} \quad r_x = r_z = 2000\,\mathrm{m} \tag{43}$$

$$L = \sqrt{\left(\frac{x - x_\mathrm{c}}{r_x}\right)^2 + \left(\frac{z - z_\mathrm{c}}{r_z}\right)^2} \tag{44}$$

$$\theta' = -2\cos^2\left(\frac{\pi L}{2}\right) \tag{45}$$

The results shown in Fig. 4 are quite similar to the ones of the dry heat bubble, except the final solution shows a slight asymmetry.

5.4 DYCOMS-II

To test the complete microphysics including selfcollection, autoconversion, sedimentation and evaporation of rain, as well as number concentration dependencies, the evolution of drizzling stratocumulus cloud layers are a challenging and interesting subject for research. Also the long time evolution of such cloud fields is very sensitive to the used microphysics, which makes them a good test case.

The boundary conditions for these simulations are the same as described in Wang and Feingold (2009), based on

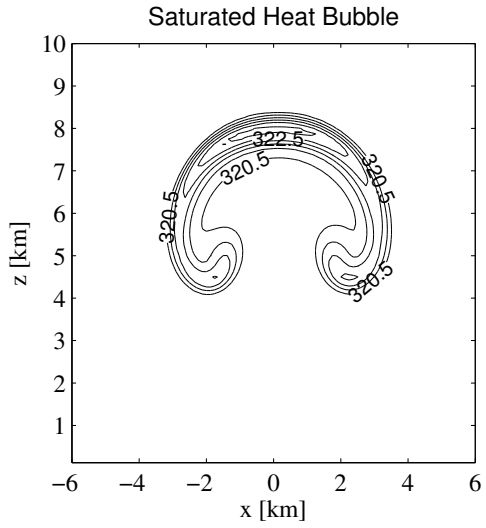

Fig. 4. Result for the moist heat bubble test case after 1000 s (contours: equiv. pot. temp. 0.5 K)

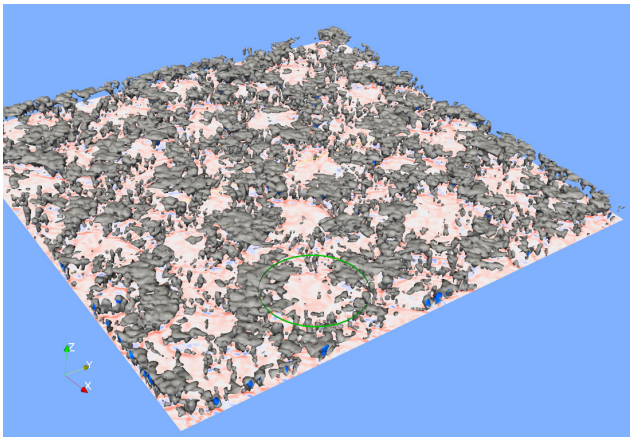

Fig. 5. DYCOMS-II: Cloud field visualisation after 9 h for the N65 case, the green circle shows the dimension of one cloud cell and has a radius of 6.5 km

measurements from the second research flight (RF2) during the Second Dynamics and Chemistry of Marine Stratocumulus field study (DYCOMS II) over the Pacific Ocean near the coast of California. One very interesting dynamic feature in such stratocumulus cloud layers are so called pockets of open cells (POCs). These are more or less cloud free areas surrounded by walls of shallow convective clouds (open cells) embedded in an elsewhere closed stratocumulus layer. The formation of those POCs seems to be coupled to the occurrence of drizzle (van Zanten et al., 2005).

In this context two simulations were performed to study the influence of cloud condensation nuclei density. To be able to compare the results to Wang and Feingold (2009) the

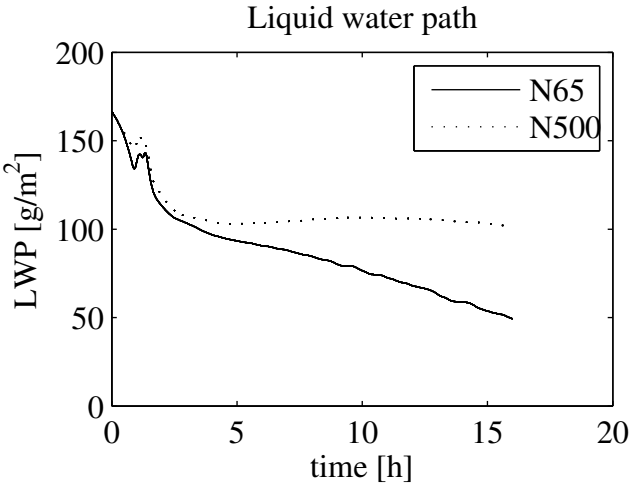

Fig. 6. DYCOMS-II: domain averaged liquid water path for the Simulations with 65 mg^{-1} (N65) and 500 mg^{-1} (N500) initialized cloud condensation nuclei density.

first example used a concentration of 65 mg^{-1} (equivalent to units of cm^{-3} when air density is 1 kg m^{-3}) and in the second run 500 mg^{-1} was used. The higher concentration produces a larger number of smaller cloud droplets, so autoconversion and drizzle formation is reduced. The result is a closed stratocumulus layer. In contrast the low concentration case produces more drizzle and the evaporation in the lower boundary layer then leads to the formation of cloud free areas surrounded by drizzling cloud walls, the open cells (see Fig. 5). In Fig. 6 the domain averaged liquid water path is shown. It decreases in the prestine simulation, while remaining at higher level in the polluted case were drizzle is suppressed. These results are also in good agreement to Wang and Feingold (2009).

5.5 First performance measurements

The 16 h integration time of this test case scenario is also used for performance measurements (see Table 1). To compare the scaling of simulation time with the number of used GPUs and domain size, several simulations with different setups were performed. In all performance test cases the vertical resolution is fixed at 60 m resulting in a domain height of 1920 m (32 cells). The horizontal size ranges from 128×128 up to 832×832 cells with a horizontal cell size of 125 m. With increasing domain size more GPUs are necessary to provide the needed amount of memory. Using more GPUs for small domain sizes strongly reduce efficiency because of increased communication between the graphics devices using the PCIe bus. Per block boundary between the GPUs or period boundary three cells are needed as buffers for the adjacent domain. So the effective domain size is reduced by six cells if the domain is split over several GPUs or has a period

Table 1. Comparison of calculation times depending on domain size and number of used GPUs (AMD Radeon HD 5870) for 16 h integration time with 1 s timestep at 125 m × 125 m × 60 m resolution for the DYCOMS-II example.

eff. domain size in cells	eff.domain size in [km]	total number of gridcells	number of GPUs	calculation time
122 × 122 × 32	15.25 × 15.25 × 1.92	524 288	1	5427 s (1.5 h)
122 × 122 × 32	15.25 × 15.25 × 1.92	524 288	2	9982 s (2.8 h)
250 × 250 × 32	31.25 × 31.25 × 1.92	2 097 152	1	24630 s (6.8 h)
250 × 250 × 32	31.25 × 31.25 × 1.92	2 097 152	2	20368 s (5.7 h)
506 × 506 × 32	63.25 × 63.25 × 1.92	8 388 608	2	68388 s (19.0 h)
506 × 506 × 32	63.25 × 63.25 × 1.92	8 388 608	4	51363 s (14.3 h)
826 × 826 × 32	103.25 × 103.25 × 1.92	22 151 168	4	250856 s (69.7 h)

boundary condition in the corresponding direction, e.g. a domain of 512 × 512 × 32 cells results in an effective domain size of 500 × 500 × 32. The largest domain size is limited by the amount of available memory on one single GPU device (currently 2 GB, 416 × 416 × 32 cells per GPU). Calculations with larger domains are theoretically possible but request swapping between the GPU and main memory, resulting in unpractical calculation times. Alternatively more computational nodes like the one used here may be combined to a cluster. The bottleneck of this multi-GPU-architecture is the PCIe bus. The used explicit methods require one data exchange per neighbour block and acoustic time step. This communication overhead is the reason why using more GPUs for small domain size strongly increases computation time. For larger domains a decrease in calculation time is observable (Table 1), e.g. for 2 million gridcells two GPUs needed 82.69 % of the time one GPU needed and for 8 million gridcells the step from two to four GPUs reduced calculation time to 75.1 %.

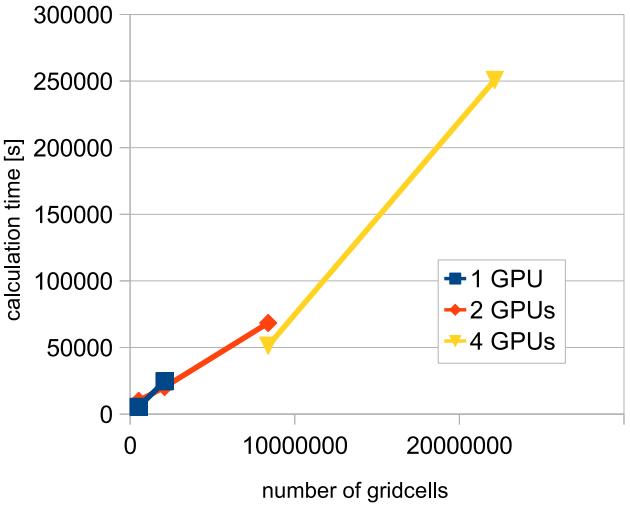

Fig. 7. First performance measurement results (details in Table 1): more GPUs may decrease performance for small domains. For larger domains speedup by using more GPUs increases with domain size.

6 Conclusions

In this work an atmospheric model based on common numeric and physical approaches is presented which uses GPUs as an efficient computation platform. Simulations for simple testcases were performed as well as more complex simulations of marine stratocumulus layers. The results are promising and compareable to simulations by other groups. This new architecture enables high resolution atmospheric modeling on small efficient devices at relative low power consumption. Even simulations with cell sizes below 100 m and domain sizes beyond 10 km × 10 km are possible in realtime. The model was already used to study the influence of an heating island surface on the marine boundary layer (Engelmann et al., 2011). Further development will include a cut cell approach to enable flows around complex obstacles as well as orographic structured landscapes like mountains and valleys. Also the study of deep convective systems including ice-phase physics are in focus of development.

Acknowledgements. S. H. thanks Deutsche Forschungsgemeinschaft for their support through grants KL 611/14 and SPP 1276 "MetStröm" and the Leibniz Gemeinschaft (WGL) for support through their PAKT program, project "Hochauflösende Modellierung von Wolken und Schwerewellen".

Edited by: A. Lauer

References

ATI Specs: http://www.amd.com/uk/products/desktop/graphics/ati-radeon-hd-5000/hd-5870/Pages/ati-radeon-hd-5870-specifications.aspx, 2009.

Bryan, G. H. and Fritsch, J. M.: A Benchmark Simulation for Moist Nonhydrostatic Numerical Models, Mon. Wea. Rev., 130, 2917–2928, 2002.

Engelmann, R., Ansmann, A., Horn, S., Seifert, P., Althausen, D., Tesche, M., Esselborn, M., Fruntke, J., Lieke, K., Freudenthaler,

V., and Gross, S.: Doppler lidar studies of heat island effects on vertical mixing of aerosols during SAMUM2, Tellus B, 63, 448–458, doi:10.1111/j.1600-0889.2011.00552.x, 2011.

Gropp, W., Lusk, E., and Skjellum, A.: Using MPI: Portable Parallel Programming with the Message-Passing Interface, MIT Press, Book, 371 pp., 1999.

Jebens, S., Knoth, O., and Weiner, R.: Explicit Two-Step Peer Methods for the Compressible Euler Equations, Mon. Wea. Rev., 137, 2380–2392, 2009.

Michalakes, J. and Vachharajani, M.: GPU Acceleration of Numerical Weather Prediction, Parallel Processing Letters, 18, 4, World Scientific, 531–548, December 2008.

Schalkwijk, J., Griffith, E., Post, F., and Jonker, H. J. J.: High performance simulations of turbulent clouds on a desktop PC: exploiting the GPU, B. Am. Meteorol. Soc., accepted, 2012.

Skamarock, W. C., Dudhia, J., Gill, D., Barker, D., Wei, W., and Powers, J.: A description of the advanced research WRF version 2. NCAR Tech. Note NCAR/TN-468+STR, National Center for Atmospheric Research, Boulder, CO, 88 pp., 2005.

Seifert, A. and Beheng, K. D.: A two-moment cloud microphysics parameterization for mixed-phase clouds, Part 1: Model description, Meteorol. Atmos. Phys., 92, 45–66, 2005.

van Zanten, M. C., Stevens, B., Vali, G., and Lenschow, D. H.: Observations of Drizzle in Nocturnal Marine Stratocumulus, J. Atmos. Sci., 62, 88–106, 2005.

Wicker, L. J. and Skamarock, W. C.: A Time-Splitting Scheme for the Elastic Equations Incorporating Second-Order Runge-Kutta Time Differencing, Mon. Wea. Rev., 126, 1992–1999, 1998.

Wicker, L. J. and Skamarock, W. C.: Time-Splitting Methods for Elastic Models Using Forward Time Schemes, Mon. Wea. Rev., 130, 2088–2097, 2002.

Wang, H. and Feingold, G.: Modeling Mesoscale Cellular Structures and Drizzle in Marine Stratocumulus. Part I: Impact of Drizzle on the Formation and Evolution of Open Cells, J. Atmos. Sci., 66, 3237–3256, 2009.