

# Evaluating Intel’s Many Integrated Core Architecture for Climate Science

Theron Voran\*, Jose Garcia†, and Henry Tufo\*†

\**University of Colorado at Boulder  
Boulder, CO 80309*

†*National Center for Atmospheric Research  
Boulder, CO 80305*

*Email: theron.voran@colorado.edu,  
jgarcia@ucar.edu, tufo@cs.colorado.edu*

**Abstract**—We evaluate Intel’s Many Integrated Core (MIC) architecture for climate science. We present preliminary performance results with a general circulation model (HOMME) and a cloud-resolving physics model (CRM). The results show promise in parallel scalability. However, single-thread performance needs further improvement.

**Keywords**—multi-core; many-core; HPC; parallel computing; climate; geoscience

## I. INTRODUCTION

With simulation now well established as the third pillar of scientific investigation alongside theory and experimentation, the field of computational science has demonstrated an almost insatiable need for large-scale computing instruments. Before 2005, the increase in raw “power” of these instruments was largely driven by an increase in single thread performance. This golden age of Moore’s law, where transistor density *and* frequency doubled roughly every 18 months, has, unfortunately, run its course [1], [2]. Future improvements will critically depend on the increased use of parallelism in all its forms, with much of the development activity focused at the node level, along with continued algorithmic improvements [3].

In response to this frequency stall, two architectural trends have recently established themselves. At the socket level the response has been to (slowly) move to multi-core and many-core architectures. The other trend has been to augment the sockets with accelerators, primarily those developed by NVIDIA [4]. Both trends have put pressure on scientific application developers who now have to manage these new parallel outlets. Typically this involves rewriting or heavily modifying existing code to fit new programming models. On the many-core side this involves mixing a threading model with message passing. In the accelerator space, the challenges include rewriting code in new languages developed for fine-grained parallelism (CUDA, OpenCL, etc.). Since accelerators have a disjoint memory space from the main CPU, data transfer becomes an issue (i.e. how often data needs to be passed between GPGPU to CPU, how much

data can fit on a GPGPU, asynchronous/overlapping data transfer, etc.) While both trends hold promise of (vastly) increased throughput, it remains to be seen precisely how many application areas will benefit.

Climate science is one of the primary drivers of simulation science as the models have an (almost) inexhaustible need for computing power [5] and the questions being addressed have immediate, wide, and deep impact to mankind. The models developed by the climate science community have had reasonable success on parallel architectures composed of shared memory nodes connected with a high-bandwidth low-latency interconnect. For example, scaling to 86K cores has been demonstrated by the High-Order Method Modeling Environment (HOMME) [6], and the Community Earth System Model (CESM) [7] can now utilize O(200K) cores [8] for coupled climate simulations.

However, the move to multi-core nodes can pose challenges to MPI-based codes, even those which have scaled well on previous systems. With the increase in core-count, off-chip memory bandwidth can become a bottleneck with all cores fighting for the same memory space. Indeed, HOMME already exhibits this behavior on current multi-core systems [9]. Another issue with increasing core-count is the amount of memory per core often decreases, either for physical density or economic reasons.

One promising recent development that has the potential to impact both trends is the introduction of Intel’s Many Integrated Core (MIC) architecture. MIC is an outgrowth of various prior research projects at Intel such as Larrabee [10] and the Teraflops chip [11]. Knights Ferry (KNF), the first generation MIC<sup>1</sup>, increases the number of cores on chip up to 32. In addition, the selection of x86 and its associated tool chain should smooth some of the programming issues. Combined, these design points hold promise for speeding application development, improving intra-node scalability, and improving performance in both homogeneous and heterogeneous node configs.

<sup>1</sup>KNF is a software development platform using alpha software and tools.

Here we present a preliminary evaluation of the KNF architecture for climate science. We present preliminary HOMME performance results on KNF with general circulation model the High-Order Method Modeling Environment (HOMME), and the Cloud-Resolving Model (CRM). The results show promise in parallel scalability, with a need for improvement in single-thread performance. We also discuss performance and scalability issues on Xeon, and possible solutions.

The rest of this paper is organized as follows. An overview of the KNF architecture is presented in section II. In section III we discuss the climate codes used in our evaluation, along with their associated challenges. Section IV describes implementation and performance results, with section V presenting conclusions and future work.

## II. KNF MIC OVERVIEW

In the following subsections we provide an overview of KNF, the first generation of the Intel MIC architecture, which is similar to an accelerator while retaining x86 characteristics.

### A. KNF Hardware

KNF is implemented on an x16 PCIe 2.0 card plugged into a Xeon host system. The KNF card has up to 32 cores clocked up to 1.2 GHz, supporting 4 hardware threads per core and a short in-order pipeline. Each core has a 512-bit SIMD vector processing unit.

Each core is provided a 32KB L1 data cache and a 32KB L1 instruction cache, as well as a 256KB L2 cache. The L2 caches for each core are interconnected via a bidirectional ring bus, creating an 8MB globally-shared fast cache. Additionally the KNF cores share 1 or 2GB of GDDR5 main memory.

Communication between a KNF card and host is performed over the PCIe bus. The KNF driver provides a virtualized Ethernet interface for administrative work, filesystem access, etc. High-performance communication is performed via a custom protocol that plugs into the OFED stack, providing a native Infiniband (IB) interface. Presumably this indicates that in the future the KNF/MIC cores will be able to communicate directly with the outside world via IB-based interconnect topologies.

### B. KNF Software

From the user perspective, KNF supports programming models which are more conventional than other accelerator platforms. Variants of the Intel compilers (icc, ifort, etc.) are provided for C/C++/Fortran. Options for parallelism include OpenMP, pthreads, and MPI, along with Intel's Cilk Plus and Threading Building Blocks (TBB). All compiling is performed on the host system. KNF also supports Intel's tools for debugging and profiling (e.g. VTune and idb).

A trimmed-down Linux OS environment runs directly on the KNF card. Users are able to login directly and launch applications, mount filesystems, transfer files, etc. One KNF core is dedicated to OS operations, and by default, application threads are not assigned to this core unless the system is fully-subscribed.

There are two options for running code on KNF, either offloading from the host or standalone on the card. The offload model is similar to the GPGPU model of computation, where the code execution begins on the host CPU, and chunks of the code are then offloaded to one or more local KNF cards, incurring associated data transfer costs over the PCIe bus. Data transfer and offloading are controlled via OpenMP-style directives in the source code.

The other option is to execute code on the KNF card as a standalone host. Users may login directly to the Linux runtime on the card and launch applications, or use the MPI runtime from the host to launch executables on the card. Currently MPI codes may be run within the KNF card or between the KNF card and the host, though more flexibility is expected in the future.

As mentioned above, KNF differs from previous accelerator architectures in several key ways. KNF has a Linux OS runtime on the card itself. KNF is based on the x86 architecture, which eases the porting of existing code, due to the availability of familiar programming models such as OpenMP and MPI. However, these programming models do not currently provide as much control over data locality within a system as do OpenCL or CUDA.

## III. CHALLENGES OF THE NEXT GENERATION CLIMATE MODELS

Modeling the climate with increased fidelity has long been a goal of the scientific community. More than ever, in order for policy makers, communities, and society in general to understand the impacts of climate change and variability, it is paramount to be able to make predictions of future climate with a much higher degree of confidence. One area of climate modeling ripe for improvement is the modeling of the Earth's atmosphere. We briefly review atmospheric modeling fundamentals and focus on two aspects that can potentially be dramatically improved with increased computational power: increased numerical resolution and improved representation of cloud processes.

At its core, a global model describes the large-scale circulation of air on the planet along with the physical properties of the fluid. The equations for the phenomena addressed by the model are stated in the space continuum and advanced in time. These equations typically have no known analytical solution, and are numerically approximated. The numerical methods discretize the continuum using a mesh or grid, for which the approximate solution is computed at a set of given points. For example, figure 1 shows a quasi-uniform discretization of the Earth using a finite element mesh

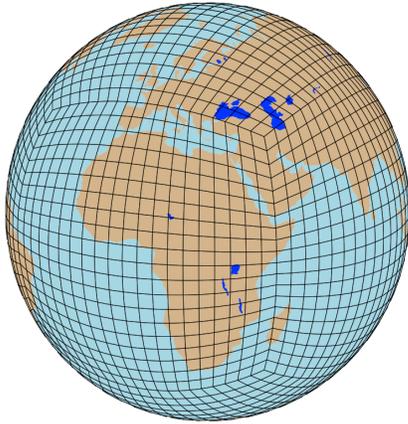


Figure 1. The Earth discretized by an uniform Cube-Sphere of 400 elements per face.

composed of quadrilaterals. Inside each element, depending on the mathematical approximation of choice, a series of points are selected in which the variables of interest such as velocities, pressures, temperatures, etc. are to be computed.

The space of options to improve model fidelity is multidimensional: improving the quality of existing science through better parameterizations, increasing ensemble sizes, decreasing time step size, improving spatial resolution, and improving model completeness by adding new science to describe additional phenomena. Our current focus is employing higher resolution coupled with explicit formulation of phenomena previously parameterized. Stan et al. [12] recently demonstrated the promise of such an approach using a coupled ocean-atmosphere climate simulation.

Once we reach a spatial discretization of approximately 10 km, we reach a point in the energy spectrum where clouds become relevant, and explicit representation of their dynamics is required. This introduces a new costly component in the full atmospheric model. What used to be computed at lower resolution by means of a simple function evaluation (parameterization) now becomes a full computational model requiring one to two orders of magnitude increase in computational cost. Thus, increased model resolution combined with a sub-grid scale cloud resolving model, dramatically increase the computational demands of the atmospheric component of a coupled climate simulation.

The computational effort required by this approach seems to be a good fit for emerging computing platforms, such as multi-core, many-core, and accelerators. However, the process of porting the required climate model components to such systems is not a trivial task. While in the past the usage of standards for parallelism representation, such as MPI, have sufficed for the model to perform well in every new system conforming to the standard, one can't rely anymore on the compiler to produce a program that performs well for

the target hybrid platforms.

#### A. HOMME

The High-Order Method Modeling Environment (HOMME) is a spectral element atmospheric general circulation model used to resolve in the discrete domain the so-called primitive equations under the hydrostatic hypothesis. Often referred to as a dynamical core, HOMME is the component of an atmospheric model that evolves the fluid dynamics. HOMME is mass and energy conserving [13], relies on a spatial discretization of the Earth as represented in a cube-sphere topology, supports both h- and p-refinement, and seeks the best local approximation, in a variational sense, to the solution of the system of PDEs for each element. HOMME is written in modular Fortran 90. Spectral element methods have been shown to scale exceedingly well on a wide range of applications. This is primarily due to the fact that only  $C^0$  continuity is required at element boundaries, and local operations can be expressed as matrix-vector or matrix-matrix products, which typically perform well on microprocessor-based architectures due to the increased floating-point operations per byte transfer. HOMME is no exception, it has scaled well on a variety of architectures, from traditional Linux clusters to MPPs like IBM's Blue Gene [14] [15].

#### B. CRM

The Cloud Resolving Model (CRM) describes the phenomena of convective clouds and encompass three fundamental physical processes: fluid dynamics, the microphysics of the hydrological cycle (ice, graupel, rain, snow, etc.), and energy absorption and radiation. The fluid dynamics portion is generally the dominant component.

From the fluid dynamics perspective, the dynamics of convective clouds are best described by the compressible Navier-Stokes equations, which are difficult to solve numerically due to the presence of sound waves. As such, models usually assume a set simplifying hypotheses, such as the hydrostatic approximation, incompressible Navier-Stokes or the Anelastic approximation [16].

Even with a simplified model description, when the Reynolds number is large, many small eddies are present in the system. Under these circumstances, if one were to obtain an explicit solution for the underlying equations, all persistent eddies must be resolved. This method, called Direct Numerical Simulation (DNS), is not viable because the required computational power is well beyond what is currently obtainable using the best contemporary supercomputers in the world, and this will remain true for the foreseeable future.

CRM instead uses an alternative approach known as a Large Eddy Simulation (LES) [17] [18]. LES is based in the theory of self similarity, primarily developed by Kolmogorov, which states that large scale eddies are geometry

dependent, while small scale eddies are more universal. Hence, LES focuses on solving the large eddies where most of the energy is, and resolve smaller scale eddies by sub-grid averaging. As such, LES simulations apply a low-pass filter to eliminate the small scale of the solution and let the computational work focus on the large scale eddies present in the dynamics. This greatly reduces the total cost of the computation and, given sufficient resolution, resolves the portion of the spectrum of interest.

From a computer science perspective, CRM is much more amenable to emerging architectures where the cost of data movement is at a premium. Because, in contrast to HOMME, CRM is an embarrassingly parallel model that has an extremely high ratio of computation to communication. This is due to the parallel decomposition for CRM, where the physical domain is simply divided into independent volumes in which the dynamics are performed independently. Inside each volume of air, the advection dominates the total cost of execution. Numerically, this phenomena is an implementation of the Multidimensional Positive Definite Advection Transport Algorithm (MPDATA) [19] [20], which is essentially a second-order accurate, positive definite, conservative finite difference solver for geophysical flows. MPDATA can represent up to 70% of the total cost of execution of the model, depending on the size of the domain and the granularity in which the domain is partitioned.

In summary, CRM in the context of climate modeling can be best envisioned as a model within the model. Its parallel characteristics, its floating-point cost, and its potential to deliver better science make it a promising approach for multi-core, many-core, and accelerated architectures.

#### IV. RESULTS

We test HOMME and CRM on a single KNF card and the Xeon host system. The host is a two-socket 3.33 GHz Intel Xeon Westmere system, each with 6 cores for a total of 12 cores. Hyper-threading was disabled on Xeon. Testing was performed with OpenMP, MPI, and hybrid parallel modes for HOMME. CRM currently only supports MPI for parallelism. The latest cross-compiler for KNF and Xeon was used, along with the Intel MPI library from the KNF software stack.

##### A. HOMME

HOMME was configured to run 2 days of the baroclinic test case, with a total of 486 elements ( $9^2$  elements per cube face, 6 faces, 20 vertical levels). The model was compiled with both MPI and OpenMP support, and run in single-precision mode. Vectorization via compiler flags was disabled for KNF because of poor performance with vectorization enabled<sup>2</sup>. The total amount of work is held constant while increasing the thread count (strong scaling).

<sup>2</sup>HOMME performs slower on KNF with vectorization enabled than with vectorization disabled. The KNF compilers are not yet capable of reliably extracting/scheduling the parallelism in HOMME.

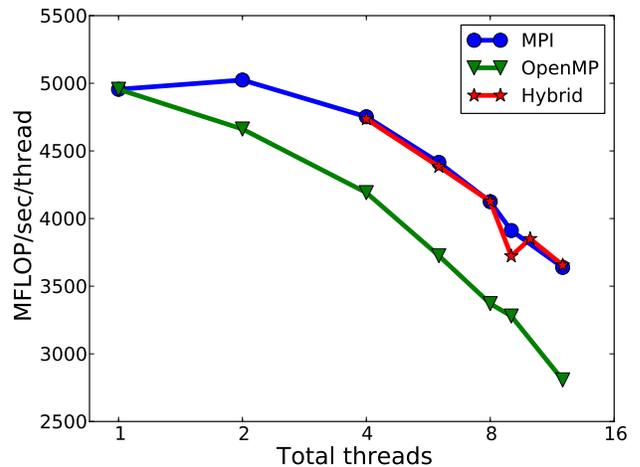


Figure 2. Sustained MFlops per thread for the HOMME baroclinic test case on Xeon. Explicit integration,  $6 \times 9^2$  elements, 20 vertical levels.  $\Delta t = 60$  sec, NP=8. MPI, OpenMP, and hybrid modes are shown. Hyper-threading was disabled on Xeon.

PAPI on a separate system was used to estimate the total number of flops in the simulation. File I/O for the model is disabled during the tests.

Sustained MFlop/sec is shown in figure 2 for Xeon, showing MPI, OpenMP, and hybrid configurations. HOMME on the host Xeon scales rather poorly, with sustained performance starting to drop at 4 threads. The MPI and hybrid configurations perform better than the OpenMP configuration. The single-thread performance on the Xeon system is a little over 18% of peak, and drops to 10% at 12 threads. We suspect memory bandwidth contention is at least partly to blame for the decrease in single-thread performance.

On KNF the code scales well, though single-thread performance is poor. Since absolute performance and comparison data cannot be released, KNF scaling plots are not included here as they would have little value.

##### B. CRM

CRM was configured to run the GATE test case, with a domain of  $128 \times 128 \times 64$ , for 120 time steps. The model was built with MPI for parallelism. As with HOMME, strong scaling was employed. VTune on the host system was used to estimate the total number of flops in the simulation. File I/O time was subtracted from the run time.

Figure 3 shows sustained MFlop/sec performance for CRM on Xeon. CRM on the Xeon system starts at 19% of peak, and drops to 11% at 8 threads<sup>3</sup>. Initial analysis using hardware performance counters on Xeon suggests that while the top routine (advection) is performing well, some ancillary routines are performing poorly and therefore

<sup>3</sup>CRM scaled only to 8 threads on Xeon because of domain decomposition constraints (powers of two).

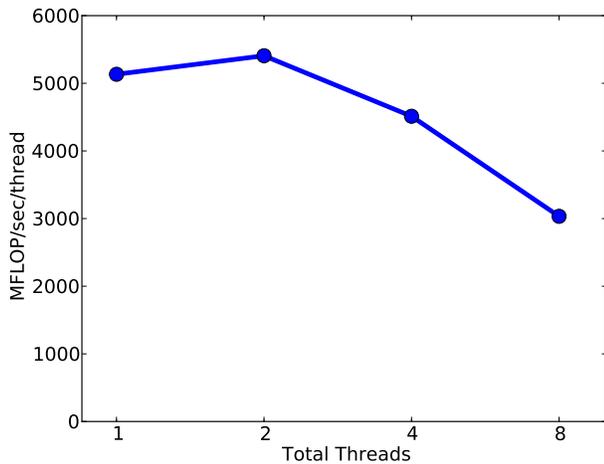


Figure 3. Sustained MFlops per thread for the CRM GATE test case on Xeon. Domain size  $128 \times 128 \times 64$ , 120 time steps. MPI used for parallelism and hyper-threading is disabled on Xeon.

reduce aggregate performance. Some of those routines are hardcoded to use double-precision, which we suspect is contributing to the poor performance.

Again, on the code scales well on KNF, though single-thread performance is poor. Since absolute performance data cannot be released, KNF scaling plots would have little value and are not included here.

## V. CONCLUSIONS AND FUTURE WORK

We evaluated the MIC architecture’s suitability for climate modeling using HOMME and CRM on KNF. In general, we observed excellent scalability with both models on KNF. However, single-thread performance is poor with or without vectorization enabled, which may indeed be the cause of the exceedingly favorable scalability. If vectorization can be fixed, it will be interesting to see how the scalability of these codes on MIC will be affected. On Xeon, we see that single-thread performance is decent, with scalability dropping off. This is indicative of the performance problems expected with these codes when moving to many-core systems.

The relative ease in porting the full models to KNF provides hope that as the hardware and software progress, it will become possible to achieve good computational efficiency. We expect that the computational rate of the individual MIC cores will be slightly lower than the Xeon cores, but in aggregate the MIC system should outpace the Xeon host. Our metrics for comparison will be performance of one Xeon socket to one KNF card/socket along with throughput per watt.

As the MIC architecture progresses, we plan to explore different strategies for the best utilization of the platform. Depending on the problem size, it may be advantageous to map the models to different parts of the system, such

as placing CRM on the MIC portion and HOMME on the host system. We also plan to evaluate MIC in a cluster configuration and explore different programming models on the platform.

## ACKNOWLEDGMENT

We would like to thank our collaborators at Intel and Intel Research for providing hardware and guidance. John Dennis at NCAR provided valuable insight into the characteristics of the models used in this investigation. NCAR is sponsored by the National Science Foundation.

## REFERENCES

- [1] K. Olukotun and L. Hammond, “The future of microprocessors,” *Queue*, vol. 3, pp. 26–29, September 2005. [Online]. Available: <http://doi.acm.org/10.1145/1095408.1095418>
- [2] M. Snir, W. Gropp, and P. Kogge, “Exascale research: Preparing for the post-moore era,” University of Illinois, Tech. Rep., June 2011. [Online]. Available: <http://hdl.handle.net/2142/25468>
- [3] “A science-based case for large-scale simulation.” Office of Science, U.S. DOE, Tech. Rep., 2003. [Online]. Available: <http://www.pnl.gov/scales/>
- [4] P. M. Kogge and T. J. Dysart, “Using the top500 to trace and project technology and architecture trends,” in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC ’11. New York, NY, USA: ACM, 2011, pp. 28:1–28:11. [Online]. Available: <http://doi.acm.org/10.1145/2063384.2063421>
- [5] T. W. Group and A. H. C. for a Petascale Collaboratory for the Geosciences., “Establishing a petascale collaboratory for the geosciences: Technical and budgetary prospectus.” UCAR/JOSS, Tech. Rep., 2005.
- [6] J. M. Dennis, J. Edwards, K. J. Evans, O. Guba, P. H. Lauritzen, A. A. Mirin, A. St-Cyr, M. A. Taylor, and P. H. Worley, “Cam-se: A scalable spectral element dynamical core for the community atmosphere model,” *International Journal of High Performance Computing Applications*, vol. 26, no. 1, pp. 74–89, 2012. [Online]. Available: <http://hpc.sagepub.com/content/26/1/74.abstract>
- [7] Community earth system model (cesm). [Online]. Available: <http://www.cesm.ucar.edu/>
- [8] P. Worley, A. Craig, J. Dennis, A. Mirin, M. Taylor, and M. Vertenstein, “Performance of the community earth system model,” in *High Performance Computing, Networking, Storage and Analysis (SC), 2011 International Conference for*, nov. 2011, pp. 1–11.
- [9] J. Diamond, B. Kim, M. Burtscher, S. Keckler, K. Pingali, and J. Browne, “Multicore optimization for ranger,” in *2009 TeraGrid Conference*, 2009.

- [10] L. Seiler, D. Carmean, E. Sprangle, T. Forsyth, M. Abrash, P. Dubey, S. Junkins, A. Lake, J. Sugerma, R. Cavin, R. Espasa, E. Grochowski, T. Juan, and P. Hanrahan, "Larrabee: a many-core x86 architecture for visual computing," *ACM Trans. Graph.*, vol. 27, pp. 18:1–18:15, August 2008. [Online]. Available: <http://doi.acm.org/10.1145/1360612.1360617>
- [11] T. G. Mattson, R. Van der Wijngaart, and M. Frumkin, "Programming the intel 80-core network-on-a-chip terascale processor," in *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, ser. SC '08. Piscataway, NJ, USA: IEEE Press, 2008, pp. 38:1–38:11. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1413370.1413409>
- [12] C. Stan, M. Khairoutdinov, C. A. DeMott, V. Krishnamurthy, D. M. Straus, D. A. Randall, J. L. Kinter, and J. Shukla, "An ocean-atmosphere climate simulation with an embedded cloud resolving model," *Geophysical Research Letters*, vol. 37, no. 1, Jan. 2010. [Online]. Available: <http://dx.doi.org/10.1029/2009GL040822>
- [13] M. A. Taylor, J. Edwards, S. Thomas, and R. Nair, "A mass and energy conserving spectral element atmospheric dynamical core on the cubed-sphere grid," *Journal of Physics: Conference Series*, vol. 78, no. 1, p. 012074, 2007. [Online]. Available: <http://stacks.iop.org/1742-6596/78/i=1/a=012074>
- [14] G. Bhanot, J. M. Dennis, J. Edwards, W. Grabowski, M. Gupta, K. Jordan, R. D. Loft, J. Sexton, A. St-Cyr, S. J. Thomas, H. M. Tufo, T. Voran, R. Walkup, and A. A. Wyszogrodski, "Early experiences with the 360TF IBM Blue Gene/L platform," *International Journal of Computational Methods (IJCM)*, vol. 5, no. 2, pp. 237–253, 2008.
- [15] R. Loft, S. Thomas, and J. Dennis, "Terascale spectral element dynamical core for atmospheric general circulation models," in *Supercomputing, ACM/IEEE 2001 Conference*, nov. 2001, p. 26.
- [16] D. K. Lilly, "A comparison of incompressible, anelastic and boussinesq dynamics," *ATMOSPHERIC RESEARCH*, vol. 40, pp. 143–151, 1996.
- [17] M. F. Khairoutdinov and Y. L. Kogan, "A large-eddy simulation model with explicit microphysics: validation against aircraft observations of a stratocumulus-topped boundary layer." *J. Atmos. Sci.*, vol. 2115-2131, 1999.
- [18] P. J. Mason, "Large-eddy simulation: A critical review of the technique," *Q.J.R. Meteorol. Soc.*, vol. 120, no. 515, pp. 1–26, 1994. [Online]. Available: <http://dx.doi.org/10.1002/qj.49712051503>
- [19] P. K. Smolarkiewicz, "Multidimensional positive definite advection transport algorithm: An overview," *International Journal for Numerical Methods in Fluids*, vol. 50, no. 10, pp. 1123–1144, 2006. [Online]. Available: <http://doi.wiley.com/10.1002/ffd.1071>
- [20] 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, 1990.