Generated domain-specific sparse data kernels for high-performance Lattice Boltzmann Methods

Philipp Suffa[1], Ulrich Rüde[12]

[1]) Friedrich Alexander Universität Erlangen-Nürnberg, Department of Computer Science, Chair System Simulation, Cauerstraße 11, 91058 Erlangen, Germany
philipp.suffa@fau.de
[2]) CERFACS, 42 Avenue Gaspard Coriolis, 31057 Toulouse Cedex 1, France
ulrich.ruede@fau.de

We present an extension of the code generation pipeline for the domain-specific Lattice Boltzmann kernels within the framework *lbmpy* [3]. This extension enables *sparse* data kernels, i.e., kernels relying on indirect addressing. Unlike direct addressing kernels, where every lattice cell of the domain is stored in memory, sparse data kernels only store and compute on fluid cells, while boundary cells are omitted. This is especially effective for domains containing a large number of boundary cells, as they typically occur in porous media simulations. Since these sparse data kernels are implemented in the *lbmpy* framework, they support a large variety of collision models and stencils. Additionally, the kernels can be generated for many common CPUs as well as for NVIDIA and AMD GPUs.

Furthermore, the sparse data kernels are extended to support an in-place streaming pattern, namely, the AA-pattern [2]. This method avoids storing a temporary PDF field. Thus we save memory and reduce the number of memory accesses for the LBM kernel from $(3 \cdot q)$ to $(2 \cdot q)$ PDF accesses, where q is the number of PDFs in the stencil. Compared to direct addressing kernels, the benefit of the AA pattern for indirect addressing kernels is even greater because the index list, which is used to store neighboring information of the PDF list, is only accessed in every second time step, so the index list accesses in an AA-pattern sparse LBM kernel decreases from $q-1$ to $(q-1)/2$ [9]. This is particularly useful because large-scale simulations based on the LBM are, in general, memory-bound, so memory accesses are the performance bottleneck of this method. Therefore it is very effective to reduce the memory accesses by $1 - \frac{2 \cdot q + (q-1)/2}{3 \cdot q + q - 1} \sim 37\%$. Ideally, the performance could improve by the same factor.

These generated sparse data kernels are integrated into the massively parallel multiphysics framework waLBerla [4]. waLBerla supports a wide range of applications. In particular, it is employed in soft matter simulations with ESPResSo [8]. Here, however, we use the LAGOON [7] test case with 188 million fluid cells on 720 cores of the JUWELS cluster [1] as a prototype scenario taken from the EU HPC project SCALABLE [6] to simulate fluid flow around a prototypical landing gear. See Figure 1 showing the Q-criterion of the flow field.

Additional scaling experiments will demonstrate the strong scalability of waLBerla with the generated domain-specific sparse data kernels, including benchmarks on GPU clusters such as JUWELS booster [1] and LUMI [5]. Furthermore, the sparse data kernels will be optimized by enabling SIMD vectoriza-

tion on CPUs and communication hiding for all architectures. Weak scalability will be demonstrated with a porous media simulation, while the LAGOON test case serves to show the strong scalability of the framework.

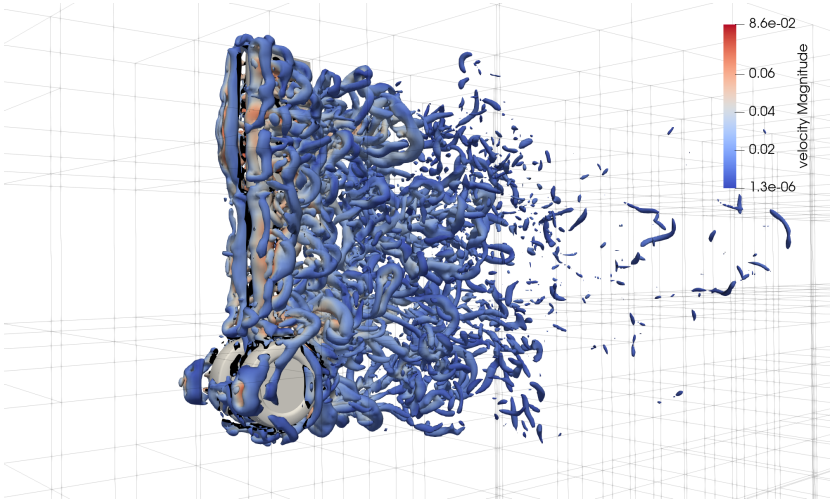

Figure 1: LAGOON test case visualized by the q-criterion. The grey lines indicate the domain partitioning into multiple waLBerla blocks.

References

[1] Alvarez, Damian, 2021. JUWELS Cluster and Booster: Exascale Pathfinder with Modular Supercomputing Architecture at Juelich Supercomputing Centre. *Journal of large-scale research facilities JLSRF*

[2] Bailey P. and Myre J. and Walsh S. and Lilja D. and Saar M. 2009. Accelerating Lattice Boltzmann Fluid Flow Simulations Using Graphics Processors. *2009 International Conference on Parallel Processing*, pp. 550-557

[3] Bauer M. and Köstler H. and Rüde U. 2021. lbmpy: Automatic code generation for efficient parallel lattice Boltzmann methods. *Journal of Computational Science*

[4] Bauer M. and Eibl S. and Godenschwager C. and Köstler H. and Rüde U. 2021. waLBerla: A block-structured high-performance framework for multiphysics simulations. *Computers Mathematics with Applications*, v. 81, pp. 478-501

[5] European HPC system LUMI. `https://www.lumi-supercomputer.eu/`

[6] SCALABLE EU project. `https://scalable-hpc.eu/`

[7] Sengissen A. and Giret J. and Coreixas C. and Boussuge J. Simulations of LAGOON landing-gear noise using Lattice Boltzmann Solver. *21st AIAA/CEAS Aeroacoustics Conference*

[8] Weik F. and Weeber R. and Szuttor K. and Breitsprecher K. and de Graaf J. and Kuron M. and Landsgesell J. and Menke H. and Sean D. and Holm C. 2019. ESPResSo 4.0 – an Extensible Software Package for Simulating Soft Matter Systems. *The European Physical Journal Special Topics*, v.227, n.14, pp.1789–1816

[9] Wittmann M. and Zeiser T. and Hager G. and Wellein G. 2013. Comparison of different propagation steps for lattice Boltzmann methods. *Computers & Mathematics with Applications*