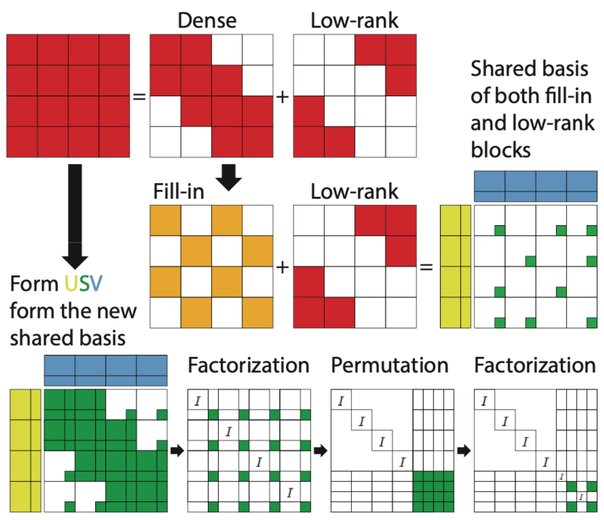
Applications on TSUBAME3.0
Deep Learning and Large Scale CFD

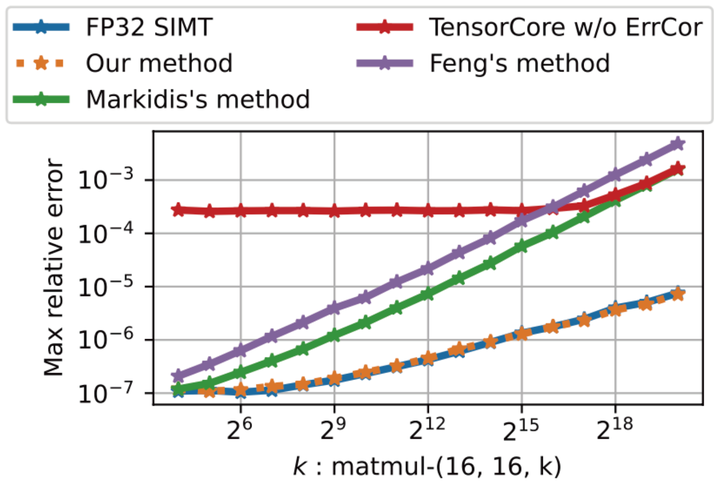
Fast Algorithms for HPC and Deep Learning

Structured Low-Rank Matrices



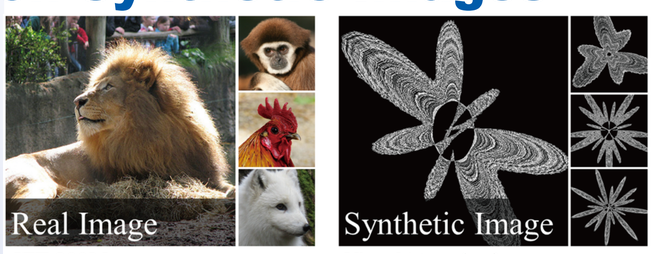
Structured low-rank matrices are dense matrices that can be hierarchically subdivided to yield low-rank off-diagonal blocks. HSS matrices are a special type of structured low-rank matrix where only the diagonal blocks are recursively subdivided. For such matrices Cholesky/LU factorization can be done without dependency on trailing submatrices. We extend this highly parallel factorization from HSS matrices to more general H^2-matrices, which can be applied to 3-D problems.

Error Correction for TensorCores



TensorCores multiply two 16bit matrices and accumulate into a 32 bit matrix. This conversion to 16bit results in a significant loss in precision. The use of an auxiliary 16bit matrix to store the mantissa loss is known to partially correct this error. However, the round-to-zero in TensorCore accumulation is another source of error that has not been accounted for in previous work. We develop a novel method that also corrects for this error, which allows us to exactly match an GEMM in single precision, while exploiting the compute capability of TensorCores.

Pretraining Vision Transformers on Synthetic Images



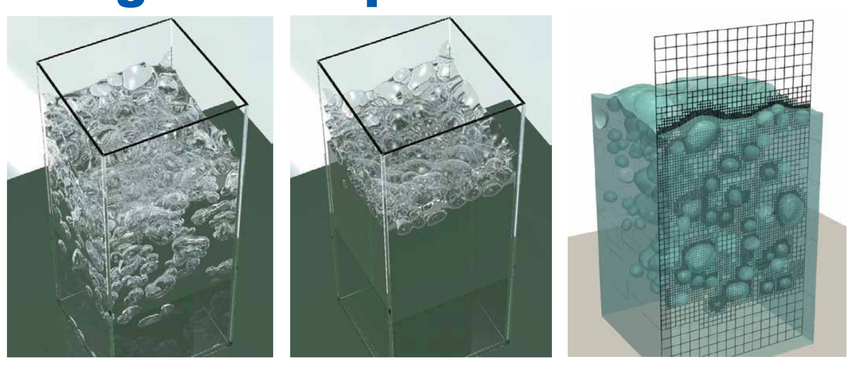
Pretraining of vision transformers requires a huge amount of real images, but huge datasets such as JFT-300M are not publicly available. In order to address this issue of data availability we investigate the possibility to train vision transformers on synthetic images generated from mathetical equations. Such images are free of copyright, privacy issues and societal bias. We are able to match the pretraining effect of real images when using our synthetic image dataset VisualAtom.

Large Language Models

We are currently involved in multiple international efforts to pretrain some of the largest language models. The systems we are using include Fugaku, LUMI, and soon Frontier and Aurora. Pretraining of these large language models requires many HPC tecniques for reducing memory consumption and I/O, while extracting the full potential of low-precision matrix engines. Optimal combination of data parallel, tensor parallel, pipeline parallel, and sharded parallel techniques are the key to achieving high throughput.

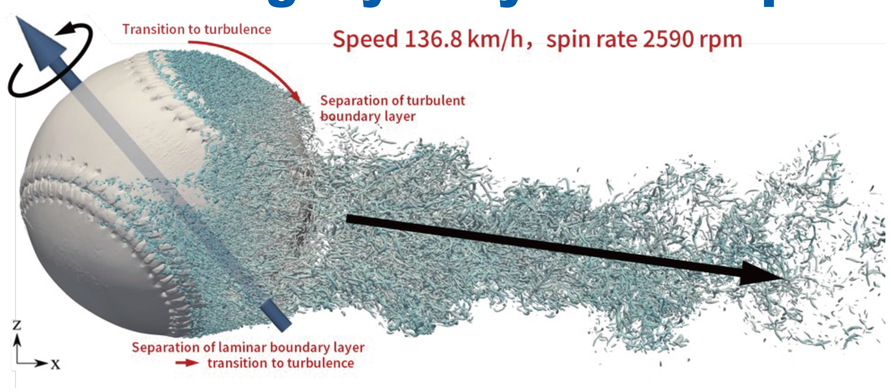
Large-scale Mesh-based and Particle-based Simulations

A Large-scale Foam Simulation using a Multi-phase Field LBM and AMR



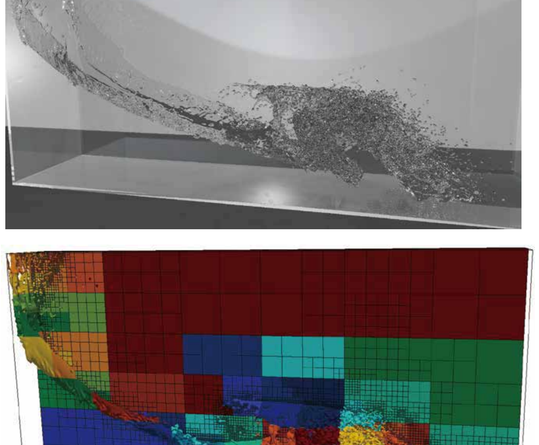
Foam formation with a stable thin liquid film is very difficult to simulate using conventional methods due to the limitation of mesh resolution that can be used. We addressed this issue by using a Multi-phase Field Lattice Boltzmann Method and Adaptive Mesh Refinement. Using the Multi-Phase Field LBM, we can prevent a “numerical coalesce” phenomenon that leads to bubble break-up. Adaptive Mesh Refinement has been introduced so that the thin liquid film can be simulated efficiently. Herein, we demonstrate a simulation of foam formed from 200 air bubbles using our proposed method.

Unraveling Mystery of Sweeper



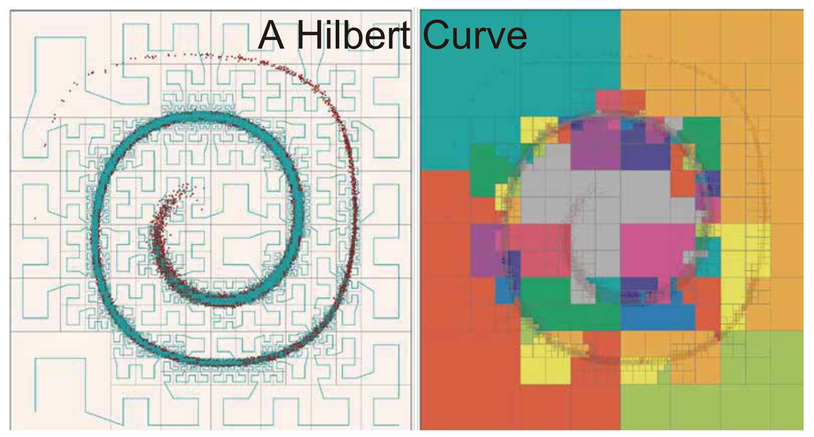
It is well known that MLB Los Angeles Angels Shohei Ohtani shows incredible performances for both batter and pitcher. He throws a kind of slider “Sweeper” (breaking ball) and its ball trajectory has more than 40cm horizontal movement but small movement in vertical direction. We study the aerodynamics of the sweeper by using a simulation based on the Lattice Boltzmann Method with a cumulant collision term suitable for a large-eddy simulation model. It is found that a lift force appears when the spin axis inclines to batter direction with 50-60 degree.

AMR for Multi-phase Flows



Simulations for multi-phase flows require high-resolution grids to capture phenomena at the interface. By using the adaptive mesh refinement (AMR) method, which dynamically adapts high-resolution grids to interfaces, computational cost and memory usage are reduced. The spatial distribution of a computational load change in time; therefore, dynamic domain partitioning using a space-filling curve is introduced for multi-GPU computing to assign an equal number of grid points to each GPU. The figures show the large-scale free-surface flow simulations for the dam-breaking process and corresponding domain decomposition for 64 GPUs.

Dynamic Load Balancing using A Space-filling Curve



For large-scale particle-based simulation and Adaptive Mesh Refinement (AMR), it is a critical issue to achieve computational load balance and equal memory usage on multiple compute nodes. A domain partitioning in terms of a space-filling curve(SFC) is one of promising candidates and it is recognized that a 1-dimensional mapping of 3-dimensional space by cutting the equal length. Due to low cost of SFC domain partitioning, it is suitable for frequent re-partitioning in the simulations of unsteady phenomena.

http://www.gsic.titech.ac.jp/sc23