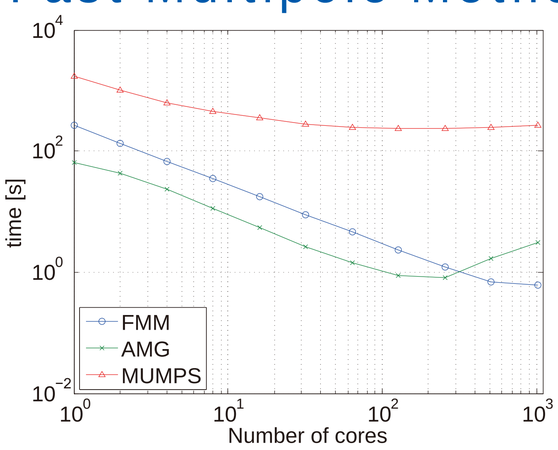
Scientific Computing in TSUBAME

Scalable Algorithms and Large-scale Simulations

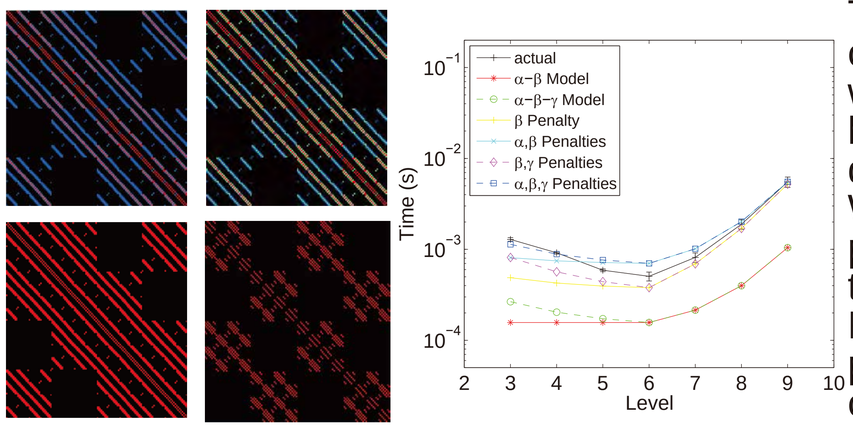
Scalable Hierarchical Algorithms for Scientific Computing

Fast Multipole Method Preconditioner



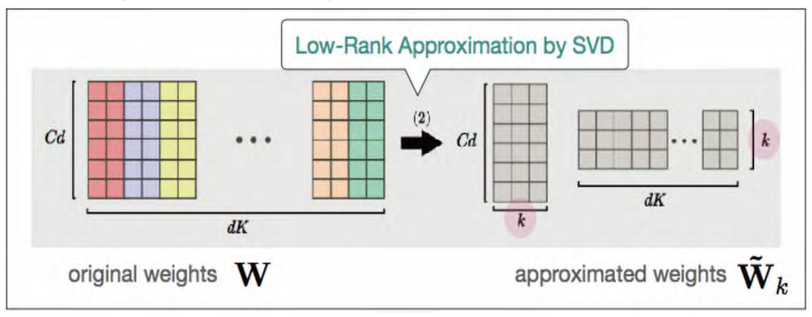
The fast multipole method (FMM) was originally developed as an O(N) algorithm for solving N-body problems. However, it can be used as a direct solver or preconditioner for solving linear systems that arise from elliptic partial differential equations that have a Green' s function solution such as Poisson' s equation. The FMM by itself can only handle far field boundary conditions, but it can be combined with boundary element methods to handle Dirichlet and Neumann boundary conditions. Comparisons against the algebraic multigrid code BoomerAMG and sparse direct solver MUMPS, shows that the FMM preconditioner becomes competitive as the degree of parallelization increases.

Performance Model for FMM Communication



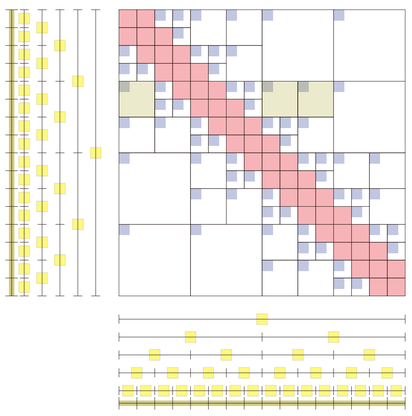
The FMM has a complex communication pattern, which results from its hierarchical and global data dependency.
We have developed a performance model for the communication in FMM, which accurately predicts the communication time for each level in the tree structure. The model considers latency, bandwidth, hops, and multicore penalties in the network. The left 4 figures show the color map of the FMM communication for different levels of the tree structure. The right figure shows the communication time at each level along with the model prediction on Shaheen2 (KAUST).

Compressing Convolutional Neural Networks



Hierarchical low-rank approximations can naturally be use in machine learning for support vector machines with kernel methods. In this work we take it one step further and use it to compress deep convolutional neural networks. The same principle as in FMM and H-matrices can be use to compress both the weights and image data. In doing so, we are able to prune the network to its optimal size without any loss of accuracy. This results in huge saving in both the memory consumption and time-to-solution.

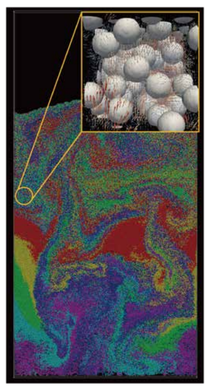
Algebraic FMM



The matrix representation of the FMM algorithm yields algebraic variants of FMM such as H-matrices and hierarchical semi-separable (HSS) matrices. These methods can approximate matrix-matrix multiplications and LU decompositions in near-linear complexity. These algebraic variants of FMM lie on the opposite end of the Byte/Flop spectrum from the analytical FMM because they store more to compute less. When the cost of data movement increases faster than arithmetic operations on future architectures, it is important to consider the whole spectrum of hierarchical low-rank approximation methods.

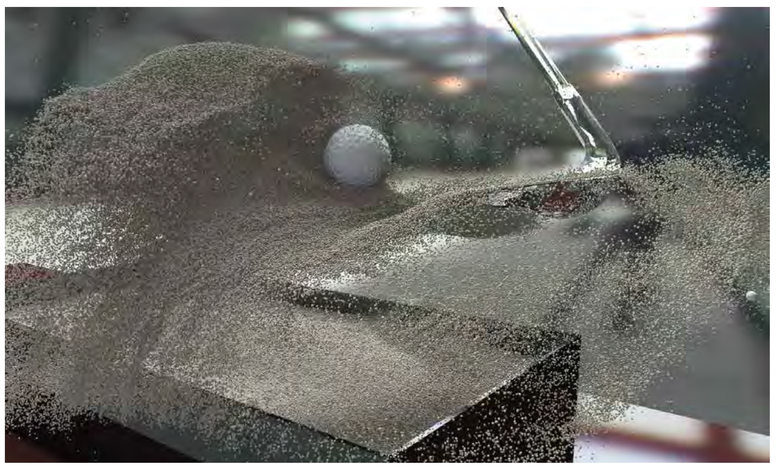
Large-scale Mesh-based and Particle-based Simulations

Gas/Liquid-Solid Two-Phase Flow Simulation



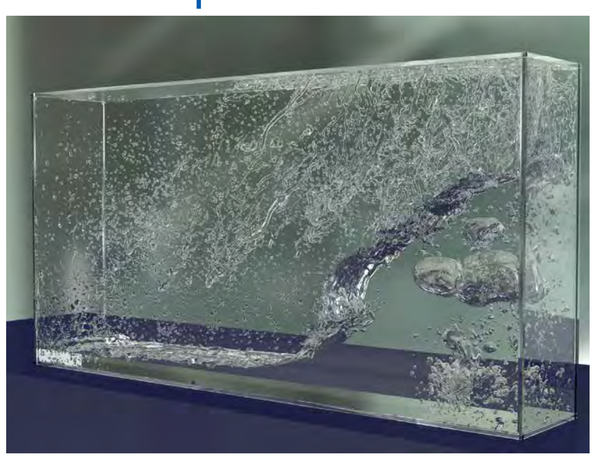
Direct interactions between fluid and solid particles are computed on the mesh to study gas/liquid-solid two-phase flows accurately for high Reynolds number flows with complex shaped particles. We have carried out a large-scale simulation of a fluid-particle system coupled lattice Boltzmann method (LBM) with discrete element method (DEM). A direct numerical simulation of fluidized bed with 562,500 DEM particles using 128 GPUs is shown in the right figure. The left figure demonstrates a result of the simulation for falling leaves using 2.1 billion mesh and 128 GPUs.

Large-scale Granular Simulation



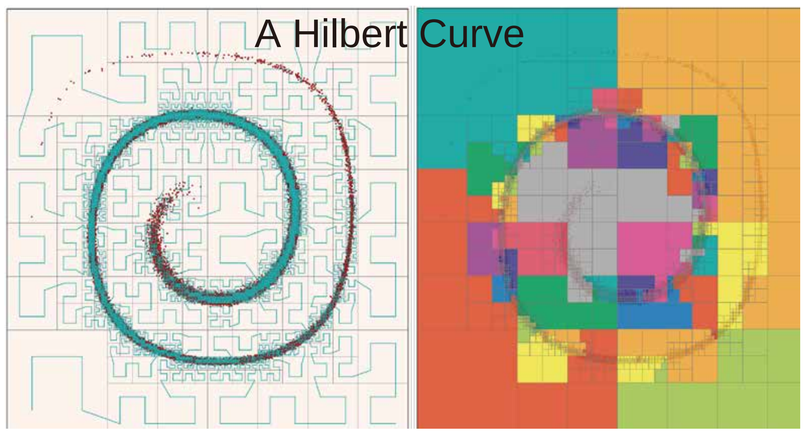
Discrete element method (DEM) is often used to simulate granular dynamics and its simple algorithm with the contact interaction is suitable for GPU computing. However so many particles are included and the particle distributions are changing in time and space. A dynamic domain decomposition has to be introduced for multiple-node computing. In a bunker shot, the sand wedge does not hit to the golf ball directly and transferring the force through the sand to the ball in order to reduce the impact. In this simulation, 16.7 millions of DEM particles are used to represent the dynamics of the sands with 256 GPUs.

Weak-Compressible Flow Computations for Gas-Liquid Two-Phase Flows



A fully explicit scheme for incompressible gas-liquid two-phase flows without solving the Poisson equation has been developed. The fractional-step method and the directional splitting method are incorporated to simplify the compressible Navior-Stokes equation to apply the method of characteristics and semi-Lagrangian method for efficient computations. The time step is determined by the sound speeds from the ideal-gas equation of state. The gas-fluid interface is captured by phase-field model and Allen-Cahn equations to reduce the volume oscillation. The results are in good agreement with those of semi-implicit incompressible flow computations.

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.