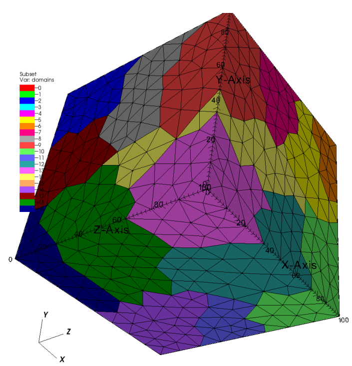
Tycho 2: A Proxy Application for Kinetic Transport Sweeps

C. K. Garrett, ckgarrett@lanl.gov
J. Warsa, warsa@lanl.gov

Tycho 2 is a proxy application that implements discrete ordinates (S_N) kinetic transport sweeps on unstructured, 3D, tetrahedral meshes. It has been designed to be small and require minimal dependencies to make collaboration and experimentation as easy as possible.

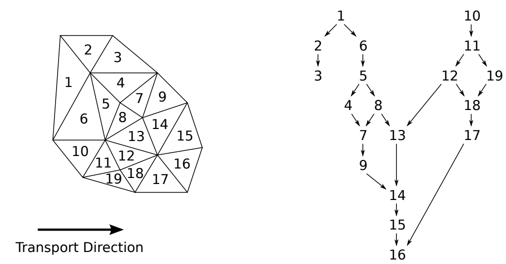


A sample partitioned, unstructured, 3D tetrahedral mesh from Tycho 2.

Background and Motivation

Kinetic transport models are used for several application areas such as: radiative transport, neutron transport, and rarefied gas dynamics. Any time implicit, kinetic code must implement an algorithm called a sweep. A sweep updates values across a mesh in an order induced by a transport direction (angle). This ordering can be described by a directed acyclic graph (DAG). An example of a 2D unstructured mesh with associated DAG is shown in the following figure.

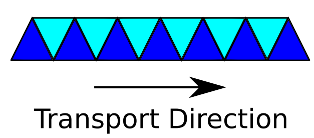
A sweep for a specific angle contains some parallelism – for instance, in the figure, cells 1 and 10 can be calculated initially at the same time. However, there is also serialization due to the fact that certain cells must be computed before other cells.



An unstructured grid with associated DAG.

A typical kinetic code will typically sweep hundreds of different angles, adding another level of parallelism.

One of the main differences between unstructured and structured grids is in their scaling properties. A structured grid can typically be partitioned such that boundaries between partitions are lines (2D) or planes (3D). Unstructured grids, on the other hand, tend to have jagged boundaries. This can require several small communications between partitions as shown in the following figure. If the light blue triangles and dark blue triangles are in different partitions, then 12 communications between the partitions of only 1 cell's information must occur to solve the sweep problem for the given direction. Such communication patterns become latency bound, which in turn reduces parallel efficiency.



Example of a mesh partitioning which requires several small communications. Different colors represent different partitions.

Description/Impact

Tycho 2 has been released as open source software [2]. The software is currently in a beta release with plans for a stable release (version 1.0) before the end of the year. The code is parallelized via MPI across spatial cells and OpenMP across angles. Currently, several parallelization algorithms are implemented.

Parallel Graph Traversal A parallel graph traversal algorithm is used as the main algorithm for sweeps parallelized across space [1, 3]. Tycho 2, with this algorithm, has already been shown to scale to thousands of processing cores as shown by the following table of weak scaling results performed on the Cielo cluster at LANL.

Cores	Time (s)	Slowdown
4	17.37	-
32	19.69	1.13x
256	22.21	1.28x
2048	29.73	1.71x
16384	35.42	2.04x

Weak scaling results on Cielo. A slowdown of 1.00x represents perfect scaling with higher values representing some inefficiency.

Parallel Block Jacobi This algorithm traverses each spatial partition's graph independently with lagged boundary data. The process is iterated until the solution does not change. At small scale, this algorithm takes longer to run than the graph traversal algorithm. However, the parallel block Jacobi algorithm is embarrassingly parallel, and hence should have better scaling properties for very large scale runs.

Schur Complement Domain Decomposition Similarly to the parallel block Jacobi algorithm, this domain decomposition uses guessed values for the incoming boundary data for each MPI rank. However, a Krylov solver is used to converge the boundary data. This should lead to faster convergence than the parallel block Jacobi method but still keep its superior scaling properties over the parallel graph traversal algorithm.

Anticipated Impact

The main goal of Tycho 2 is to be used for rapid research of both algorithmic and computational optimizations. Knowledge gained from such research will be incorporated into Capsaicin, which is a production level code at LANL that implements a discrete ordinates, kinetic transport solver on unstructured meshes. The development of Tycho 2 will also facilitate collaboration with researchers outside LANL.

Path Forward

The following list represents future goals of the project.

- Create version 1.0 release
- Add a GPU port
- Compare parallel block Jacobi and Schur complement domain decomposition to the parallel graph traversal algorithm at very large scale
- Compare Tycho 2 performance on NVIDIA GPU, XEON Phi (Knights Landing), and XEON Haswell architectures

Acknowledgements

Los Alamos Report LA-UR-16-27030. Funded by the Department of Energy at Los Alamos National Laboratory under contract DE-AC52-06NA25396. This work was carried out under the auspices of the National Nuclear Security Administration of the U.S. Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396, and supported by the Advanced Simulation and Computing Program.

References

[1] G. Colomer, R. Borrell, F. X. Trias, and I. Rodríguez. Parallel algorithms for S_n transport sweeps on unstructured meshes. *Journal of Computational Physics*, 232:118–135, 2013.

[2] C. K. Garrett. Tycho 2: Unstructured S_n sweep proxy application, 2016. https://github.com/losalamos/tycho2.

[3] Shawn D. Pautz. An algorithm for parallel S_n sweeps on unstructured meshes. *Nuclear Science and Engineering*, 140:111–136, 2002.