



Contribution ID: 62

Type: **Short talk**

## Batched Iterative Solvers in Plasma Fusion Simulations

*Tuesday 21 March 2023 16:10 (10 minutes)*

Batched linear solvers, which solve many small related but independent problems, are important in several applications. This is increasingly the case for highly parallel processors such as graphics processing units (GPUs), which need a substantial amount of work to keep them operating efficiently and solving smaller problems one-by-one is not an option. Because of the small size of each problem, the task of coming up with a parallel partitioning scheme and mapping the problem to hardware is not trivial. In recent history, significant attention has been given to batched dense linear algebra. However, there is also an interest in utilizing sparse iterative solvers in a batched form, and this presents further challenges. An example use case is found in a gyrokinetic Particle-In-Cell (PIC) code used for modeling magnetically confined fusion plasma devices. The collision operator has been identified as a bottleneck, and a proxy app has been created for facilitating optimizations and porting to GPUs. The current collision kernel linear solver does not run on the GPU—a major bottleneck. As these matrices are well-conditioned, batched iterative sparse solvers are an attractive option. A batched sparse iterative solver capability has recently been developed in the Ginkgo library. In this paper, we describe how the software architecture can be used to develop an efficient solution for the XGC collision proxy app. Comparisons for the solve times on NVIDIA V100 and A100 GPUs and AMD MI100 GPUs with one dual-socket Intel Xeon Skylake CPU node with 40 OpenMP threads are presented for matrices representative of those required in the collision kernel of XGC. The results suggest that GINKGO's batched sparse iterative solvers are well suited for efficient utilization of the GPU for this problem, and the performance portability of Ginkgo in conjunction with Kokkos (used within XGC as the heterogeneous programming model) allows seamless execution for exascale oriented heterogeneous architectures at the various leadership supercomputing facilities.

### JLESC topic

**Primary author:** ANZT, Hartwig Andreas (University of Tennessee)

**Presenter:** ANZT, Hartwig Andreas (University of Tennessee)

**Session Classification:** Short Talks on Numerical Methods

**Track Classification:** Numerical methods and algorithms