

Kokkos Kernels 4.3 Release Briefing

Kokkos Kernels team

April 25, 2024

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.
SAND2024-051850

4.3 Release Highlights

- ▶ Block ILUK
- ▶ Stream RCM re-ordering and ILUK
- ▶ Implicit time integrators
- ▶ SPMVHandle
- ▶ Miscellaneous
- ▶ Deprecations and other breaking changes

Online Resources:

- ▶ <https://github.com/kokkos>:
 - ▶ Primary Kokkos GitHub Organization
- ▶ <https://github.com/kokkos/kokkos-tutorials/wiki/Kokkos-Lecture-Series>:
 - ▶ Slides, recording and Q&A for the Full Lectures
- ▶ <https://github.com/kokkos/kokkos-kernels/wiki>:
 - ▶ Wiki including API reference
- ▶ <https://kokkosteam.slack.com>:
 - ▶ Slack channel for Kokkos.
 - ▶ Please join: fastest way to get your questions answered.
 - ▶ Can whitelist domains, or invite individual people.

Both `spiluk_symbolic` and `spiluk_numeric` now support block sparse row data.

- ▶ The block algorithms operate on the block graph of the matrix
- ▶ Operations on blocks are performed with dense linear algebra kernels
- ▶ Increasing level of fill by one adds a new block per row, i.e. block size new entries per row
- ▶ The `spiluk` handle constructor now takes a `block_size` argument defaulted to 0
- ▶ the `spiluk` handle provides two new methods:
 - ▶ `get_block_size()`
 - ▶ `set_block_size()`

Upcoming:

- ▶ new interface that accepts a `Crs` or `Bsr` matrix instead of views
- ▶ move new `Crs` interface out of experimental, keep `Bsr` in experimental at first

The utility function `kk_extract_diagonal_blocks_crsmatrix_sequential` now has an option for reverse Cuthill–McKee (RCM) re-ordering

- ▶ The function is able to
 - ▶ apply RCM re-ordering to extracted diagonal sub-matrices as indicated by the Boolean input argument `UserRCMReordering`
 - ▶ return a vector of permutation index views in RCM order corresponding to diagonal blocks if `UserRCMReordering` is true, or an empty vector if it is false
- ▶ Permutation index views have the same `memory_space` as the input CRS matrix
- ▶ Usage example:
 - ▶ `kk_extract_diagonal_blocks_crsmatrix_sequential(A_in, diagBlk_in_b);` (default - no RCM reordering)
 - ▶ `permute = kk_extract_diagonal_blocks_crsmatrix_sequential(A_in, diagBlk_out, UserRCMReordering);`
 - ▶ `diagBlk_out` and `permute` can typically be used as inputs for stream-based processing in `Ifpack2::RILUK` preconditioner

Kokkos ODE now has implicit time integrators in `KokkosODE_BDF.hpp`

- ▶ backward differentiation formulae order 1 to 5
- ▶ automatic order increase/decrease and time step increase/decrease
- ▶ automatic selection of the first time step size

Upcoming:

- ▶ implementation of Adams-Moulton implicit methods
- ▶ more uniform interface for all algorithms

Warning!

This feature is still experimental: please let us know about issues you encounter!

KokkosSparse::spmv now takes an optional handle argument.

- ▶ Stores TPL matrix analysis, allowing reuse
 - ▶ Especially benefits rocSPARSE (measured 15x speedup vs. setup every call)
- ▶ Allows algorithm selection and expert tuning (replaces Controls)
- ▶ Setup is performed automatically on the first spmv call
- ▶ `KokkosSparse::SPMVHandle<ExecSpace, AType, XType, YType> handle;`
- ▶ `KokkosSparse::spmv(space, &handle, "N", alpha, A, ...);`

Last BLAS rank update function implemented: SYR2.
It performs a symmetric rank-2 update:

$$A = A + \alpha(x * y^T + y * x^T)$$

- ▶ `KokkosBlas::syr2(space, trans, uplo, alpha, x, y, A);`
- ▶ provides: native, BLAS, cuBLAS and rocBLAS implementations
- ▶ it is thread safe and non-blocking

TPL support

- ▶ cuSOLVER and rocSOLVER libraries are added and provide vendor optimized implementation for LAPACK algorithms

Only one change this time around:

- ▶ `KokkosSparse::Experimental::spadd_symbolic` and `spadd_numeric` now require two new arguments for number of rows and columns.
- ▶ Before 4.3:
 - ▶ `spadd_symbolic(&handle, a_rowmap, a_entries, b_rowmap, ...);`
 - ▶ `spadd_numeric(&handle, a_rowmap, a_entries, a_values, ...);`
- ▶ 4.3:
 - ▶ `spadd_symbolic(&handle, m, n, a_rowmap, a_entries, b_rowmap, ...);`
 - ▶ `spadd_numeric(&handle, m, n, a_rowmap, a_entries, a_values, ...);`

Upcoming: will have a look at what is in the `Experimental` and move some kernels out of it and deprecate the experimental version.