Porting NH dycore to C++/Kokkos

The preqx target already ported and performant (GMD paper just accepted). The theta-l target porting has started.

Picture:
- several structures/routines can be recycled from the preqx effort (tracers, utilities)
- some structures need minor adjustments (MPI halo exchange, sphere operators)
- some structures need to be fully rewritten (RK stages for dynamics)

Expectations:
- have a working dycore in the next few months (3-6)
- optimize performance in the next 3 months
- hook up to e3sm and be ready for GB submission in early 2020.
Porting NH dycore to C++/Kokkos

Status:
▶ identified (and separated) code that is preqx-specific from code that can be recycled
▶ implemented “utility” ops (e.g., compute level quantities from interface quantities, or vertical derivatives)
▶ first draft of compute_andor_apply_rhs (caar) implemented

Next:
▶ test caar kernels
▶ implement HV functor (fairly similar to preqx impl)
▶ tridiag solver (check with kokkos-kernels if there’s room for recycling their work)
▶ performance optimizations