**A Barotropic Mode Solver for MPAS-Ocean using ForTrilinos**

Hyun-Gyu Kang1, Katherine J. Evans1, Philip W. Jones2, Mark R. Petersen2, Andrew G. Salinger3, Raymond S. Tuminaro4, and Andrey Prokopenko1

1*Oak Ridge National Laboratory, Oak Ridge, TN, USA*

2*Los Alamos National Laboratory, Los Alamos, NM, USA*

3*Sandia National Laboratory, Albuquerque, NM, USA*

4*Sandia National Laboratory, Livermore, CA, USA*

A semi-implicit barotropic mode solver for the Model for Prediction Across Scale-Ocean (MPAS-O) using ForTrilinos has been implemented. A nonlinear barotropic system is discretized in time with the Crank-Nicolson method to build a Helmholtz-type problem. To solve this nonlinear system by using the linear iterative solver, barotropic thickness inside each divergence operator is replaced by values from the previous time step. Therefore, the coefficient matrix is reassembled for every time step. To obtain better numerical efficiency using symmetric solvers, the barotropic system is symmetrized by multiplying the area of each cell. As a solver package to solve the system, we employ ForTrilinos which is an open-source software library providing object-oriented Fortran interfaces to Trilinos C++ packages. ForTrilinos provides Fortran interfaces for a variety of linear solvers and preconditioners through Stratimikos, IFpack2, Belos, and MueLu. In the present study, we carefully choose the algebraic multigrid preconditioner with three layers via MueLu and the conjugate gradient method with a single reduction through Belos for the preconditioner and the linear iterative solver, respectively. Computational performance using a global ocean experiment on a high-resolution global domain varying gradually from 18 km at the equator to 6 km at poles is compared with an existing explicit-subcycling scheme and the hand-coded semi-implicit solver. It is found that the Fortrilinos semi-implicit solver is faster than the current explicit scheme but slower than the hand-coded semi-implicit solver. The implementation of the semi-implicit solver using ForTrilinos is still underway so further optimizations will improve computational efficiency.