ParaSkel++: Parallel implementation of high-order Skeletal methods in C++

ParaSkel++ is a C++ platform for the high-performance, arbitrary-order, 2D/3D numerical approximation of PDEs on general polytopal meshes using skeletal Galerkin methods. It is built around the core class FiniteElement, which acts as a parent class to all particular skeletal methods implemented (such as FEM, VEM, HHO, etc). The code is essentially structured around two main parts:

1) the local driver part with the DiscreteSpace class that instantiates all the FiniteElement objects for each cell and loops over all of them to perform cell-by-cell computations;

2) the solver part with the MatrixPattern class that assembles the global matrix using a local-to-global map and the linear solver for computing the numerical solution.

This dichotomy allows for a natural parallelization in both of these parts independently. In the local driver part, we perform an "in house" parallelization by distributing the work done cell by cell among different processes. Currently, the user can switch between shared memory, distributed memory as well as hybrid parallelism modes at compile time. In the solver part, parallelization stems from what is supported by the chosen third-party solver. ParaSkel++ defaults to using linear solvers from Eigen (only shared memory mode), while allowing for the compile time option to use Intel MKL Pardiso solver (all three parallel modes). For users interested in testing the methods, the platform can also efficiently perform error norm computations against the exact solution ("in house" parallelization with all three modes). We observe good linear scaling on all "in house" parallelism and relatively optimal scaling in the parallel solvers. Finally, some work has been done for ParaSkel++ to make use of the latest developments in polytopal quadrature (implemented, in test) and for ParaSkel++ to interface a Voronoi mesher (currently under development).