diezDecomp

Cross-Platform Library for Transposes and Halo Exchanges in Extreme-Scale (DNS/LES) Simulations. Written in modern Fortran, with GPU support through OpenACC kernels and MPI.

1. Compatible with:
   - Nvidia/AMD GPU-based supercomputers (LUMI, Leonardo, Snellius, etc.)
   - CPU-based clusters (gfortran MPI, etc.)
2. Key features:
   - Supports advanced any-to-any transpose operations between mismatched 2D pencil decompositions (as shown in the `./tests` folder). Further details are explained in the paper listed below.
   - Contains a core file (`diezdecomp_core.f90`), and two API versions:
     - `diezdecomp_api_cans.f90` for compatibility with the CaNS project (https://github.com/CaNS-World/CaNS)
     - `diezdecomp_api_generic.f90` for general-purpose operations, compatible with any project.
3. Installation:
   - Please copy the source files (`diezdecomp_core.f90` and the desired API version `diezdecomp_api_*.f90`) to your work directory, and compile them together with any other Fortran code.
   - Re-run the test suite (`./tests`) and examples (`./examples`) in the target platform to ensure full compatibility.
4. Usage:
   - Please refer to the `./examples` folder for more information about using `diezDecomp` to perform halo exchanges and transpose operations.
     - The examples include both Fortran code and Jupyter notebooks with detailed explanations (for halos and transposes).
5. Working principle:
   - `diezDecomp` is flexible and robust, because it works by intersecting the `[i,j,k]` bounds of all MPI tasks in the input/output 2D pencil distributions. The results of the `[i,j,k]` intersections are internally checked, and simple transpose alternatives (like `MPI_Alltoallv`) might be accepted. However, `diezDecomp` is able to schedule `mpi_isend/mpi_irecv` pairs to handle any data communication pattern encountered.
   - For halo exchanges, both synchronous (`MPI_Sendrecv`) and asynchronous (`mpi_isend/mpi_irecv`) are available.
6. General advice:
   - Users are generally advised to use asynchronous (`mpi_isend/mpi_irecv`) modes, since they have negligible performance differences and they are much more flexible.
   - For asynchronous transpose operations (`mpi_isend/mpi_irecv`), `diezDecomp` is able to detect when information is local to each MPI task (`sender=receiver`), and it schedules a local buffer copy without expensive MPI calls.
     - This can result in massive performance improvements for 2D pencil decompositions with few partitions along one dimension (e.g. `2 x 512` to `4 x 512`).
7. Reference article:
   - Rafael Diez Sanhueza, Jurriaan Peeters, Pedro Costa (2025). A pencil-distributed finite-difference solver for extreme-scale calculations of turbulent wall flows at high Reynolds number. (https://arxiv.org/abs/2502.06296)