Geophysical Research Abstracts
Vol. 21, EGU2019-5802, 2019
EGU General Assembly 2019
© Author(s) 2019. CC Attribution 4.0 license.

A Python-based approach to the physics-dynamics coupling in atmospheric models

Stefano Ubbiali (1), Christoph Schär (2), Linda Schlemmer (2,3,4), William Sawyer (5), Thomas C. Schulthess (1,5)

(1) Insititute for Theoretical Physics, ETH Zürich, Switzerland, (2) Institute for Atmospheric and Climate Science, ETH Zürich, Switzerland, (3) Deutscher Wetterdienst, Germany, (4) Hans-Ertel Zentrum für Wetterforschung, Goethe Universität Frankfurt, Germany, (5) Swiss National Supercomputing Centre (CSCS), Switzerland

Atmospheric models are complex systems where a dynamical core solves the fluid-dynamics equations and a bundle of physical parameterizations express the bulk effect of subgrid-scale phenomena upon the large-scale dynamics. The procedure which molds all the dynamical and physical components into a coherent and comprehensive model is referred to as the *physics-dynamics coupling*. Whereas parameterizations have been largely studied in isolation, the physics-dynamics coupling has historically received less attention. To a certain extent, this deficiency may be ascribed to the lack of flexibility, interoperability, and usability of traditional frameworks.

We propose a Python framework - code named `tasmania` - to ease the composition, configuration, simulation and monitoring of Earth system models. The framework features a component-based architecture, with each component being a Python class representing a dynamical or physical process. As a result, the user is given fine-grained control on the execution flow.

Physical components must conform to `sympl`'s (System for Modelling Planets) primitives application programming interface (API). To facilitate the development of dynamical kernels, `tasmania` provides an abstract base class (ABC) with intended support for multi-stage time-integrators (e.g., Runge-Kutta schemes) and partial operator splitting techniques, which integrate slow and fast processes with large and multiple small time steps, respectively. To this end, a distinction between *slow* physics (calculated over the large time step, outside of the dynamical core), *intermediate* physics (evaluated over the large time step at every stage) and *fast* physics (computed over the shorter time step at each sub-step) is made. Four coupling mechanisms (concurrent coupling, parallel splitting, sequential-splitting, and symmetrized sequential-splitting) are currently implemented; hybrid approaches are possible.

A simplified hydrostatic model in isentropic coordinates is used as proof-of-concept. Finite difference operators arising from the numerical discretization of the model are implemented via `GridTools4Py` - a domain specific language (DSL) for stencil-based codes which offers a high-level entry point to the high-performance `GridTools` library. Early results, highlighting the variety of performance across the diverse coupling methods, are showcased.