LibPFASST x SWEET

Valentina Schüller

January 18, 2022

Abstract

This text outlines how to use the SWEET programs making use of LibPFASST. I describe central development decisions as well as some peculiarities of how these programs behave. See the SWEET tutorials for specific help with running the programs.

Contents

1 Using LibPFASST programs in SWEET

Currently supported programs:

- explicit SDC for the shallow water equations on the sphere (`libpfasst_swe_sphere_expl_sdc`)
- IMEX SDC for the shallow water equations on the sphere (`libpfasst_swe_sphere_imex_sdc`)

Not supported yet:

- MLSDC for the shallow water equations on the sphere (`libpfasst_swe_sphere_mlsdc`)

1.1 Compilation

This works:

```
scons --program=PROGRAM_STRING --quadmath=disable --libpfasst=enable
--sweet-mpi=enable --libsph=enable --plane-spectral-space=disable
--sphere-spectral-space=enable --threading=off --libfft=enable
--sphere-spectral-dealiasing=enable
```

1.2 Runtime

- so far, the programs have only been tested with the Galewsky benchmark!
- in addition to the SWEET runtime options, the LibPFASST programs have flags for customizing the PFASST algorithm. These are the important ones, use `--help` to see how to set them:
  - `libpfasst-nodes-type`: type of nodes used by LibPFASST (Gauss-Lobatto or Gauss-Legendre)
  - `libpfasst-nnodes`: number of nodes used by PFASST algorithm
  - `libpfasst-niters`: number of iterations per time step used by PFASST algorithm
  - `libpfasst-nlevels`: only relevant for MLSDC, anything above 1 will lead to an error!
- **note:** the flags `timestepping-method` and `timestepping-order` are not used by the LibPFASST programs! To compute your achievable order, see section 2

1.3 Testing

Convergence tests are contained in `tests/70_program_libpfasst_*`, compilation tests are in `tests/10_compile/10_compile_libpfasst_*`.

2 Special hints/peculiarities

2.1 ... when using the programs

- `--output-file-mode=csv` is the default for the other SWEET programs but **not supported** in the LibPFASST programs. Use `--output-file-mode=bin`
- the LibPFASST programs do not support GUI output
- in general: achievable convergence order of SDC: $\mathcal{O}\left(\Delta t^{\min(n_{\text{iters}},m)}\right)$
  - m: order of quadrature nodes – not the integration but the *interpolation* order! (1 higher than for integration)

- achievable convergence order with Gauss-Lobatto nodes

$$\mathcal{O}\left(\Delta t^{\min(n_{\text{iters}}, 2n_{\text{nodes}}-2)}\right)$$

- achievable convergence order with Gauss-Legendre nodes (for Gauss-Legendre nodes, the interval boundaries are part of the node count)

$$\mathcal{O}\left(\Delta t^{\min(n_{\text{iters}}, 2(n_{\text{nodes}}-2))}\right)$$

2.2 ... when looking at the code

- in the code, `dt` is the time step size Δt provided at runtime with the `--dt` flag. `dtq` is $\Delta t \cdot \omega$, where ω is the current quadrature weight. **Note:** for Gauss-Legendre nodes, the interval boundaries are part of the node count but their weight is of course 0, thus `dtq` can be 0!

- it might seem surprising that there exists an evaluation function for the implicit piece in the IMEX sweeper. This is intended by the IMEX-SDC algorithm in [2, Algorithm 1].

3 Development decisions

- whenever a new version of the sweeper is used, a new `libpfasst_swe_sphere_*` program should be created alongside a new compilation and convergence test

- headers shared by all LibPFASST programs are to be stored in `src/programs/libpfasst_interface`.

- the LibPFASST programs should behave as much as the `swe_sphere` program as possible, from a user's perspective. Particularly, this affects:
  - variable naming
  - output behavior
  - output file name and structure

4 More information

- LibPFASST repository and documentation

- The IMEX-SDC algorithm is described in [2, Algorithm 1]

- original SWEET x LibPFASST paper: [1]

References

[1] Francois Hamon, Martin Schreiber, and Michael Minion. Multi-Level Spectral Deferred Corrections Scheme for the Shallow Water Equations on the Rotating Sphere. *Journal of Computational Physics*, 376:435–454, January 2019. arXiv: 1805.07923.

[2] Robert Speck, Daniel Ruprecht, Matthew Emmett, Michael Minion, Matthias Bolten, and Rolf Krause. A multi-level spectral deferred correction method. *BIT Numerical Mathematics*, 55(3):843–867, September 2015. arXiv: 1307.1312.