

Chapter 3 of Wan, 2009 [1] gives a very detailed explanation on the numerics in the dycore of ICON. These further examples of operators and discretization methods provide more to think about on the expressiveness of the GridTools.

1 More evidence calling for fields on multiple location

1.1 Examples

We start from these edge-to-cell and cell-to-edge averaging operators that are used a lot in discretization defined as (the following numbering of equations refers to that in [1])

$$\bar{\varrho}_i = \sum_{l \in \mathcal{E}(i)} \frac{A_{i,l}}{A_i} \varrho_l \quad (3.13)$$

$$\check{\varrho}_l = \sum_{j=1,2} \frac{A_{i(l,j)}}{A_l} \varrho_{i(l,j)} \quad (3.14)$$

Here $i(l, 1)$ and $i(l, 2)$ denote the neighboring cells to edge l . A_l and A_i is the area associated to edge l and cell i . As A_i and A_l vary on the grid, this cannot be considered as a simple arithmetic average. Examples of (3.13) and (3.14) are

$$\text{div}(\mathbf{v} \Delta p)_{i,k} = \frac{1}{A_i} \sum_{l \in \mathcal{E}(i)} v_{n_l,k} \Delta \check{p}_{l,k} (\mathbf{N}_l \cdot \mathbf{n}_{i,l}) l \quad (3.18)$$

$$\overline{K}_{i,k} = \sum_{l \in \mathcal{E}(i)} \frac{A_{i,l}}{A_i} v_{n_l,k}^2 \quad (3.25)$$

$$\text{div}(\mathbf{v} T \Delta p)_{i,k} = \frac{1}{A_i} \sum_{l \in \mathcal{E}(i)} v_{n_l,k} \check{T}_{l,k} \Delta \check{p}_{l,k} (\mathbf{N}_l \cdot \mathbf{n}_{i,l}) l \quad (3.18)$$

Basically, we have such a form

$$\text{operator}(\varrho)_i = \sum_{l \in \mathcal{E}(i)} w_{i,l} \varrho_l$$

$$\text{operator}(\varrho)_l = \sum_{j=1,2} w_{l,i(l,j)} \varrho_{i(l,j)}$$

This more general form can also describe

$$\bar{\bar{\psi}}_l = \frac{A_{i(i,2),l}}{A_l} \psi_{i(l,1)} + \frac{A_{i(i,1),l}}{A_l} \psi_{i(l,2)} \quad (3.51)$$

A more complicated example also fits this form, considering $w_{i,l} = \frac{1}{A_i \Delta p_{i,k}} 2A_{i,l}$:

$$\left(\frac{R_d T}{p} \mathbf{v} \cdot \nabla p \right)_{i,k} = \frac{1}{A_i \Delta p_{i,k}} \sum_{l \in \mathcal{E}(i)} \left[\left(\frac{R_d T}{p} \nabla p \right)_{l,k} \cdot \mathbf{N}_l \right] u_{l,k} \Delta \check{p}_{l,k} 2A_{i,l} \quad (3.57)$$

The average divergence operator is a cell-to-cell averaging, which has the form

$$\text{operator}(\varrho)_i = \sum_{j \in \{i\} \cup \mathcal{N}(i)} w_{i,j} \varrho_j$$

where \mathcal{N}_i denote the neighboring cells to cell i .

1.2 Problems

These operators are beyond what a uniform sign or different strategies regarding coloring can describe.

2 Proposal

A proposed syntax to describe, for example,

$$\text{operator}(\varrho)_i = \sum_{l \in \mathcal{E}(i)} w_{i,l} \varrho_l$$

would be

```
struct edge_to_cell_averaging_functor {
    typedef in_accessor<0, icosahedral_topology_t::edges, extent<1> > rho;
    typedef in_accessor<1, icosahedral_topology_t::cells,
        icosahedral_topology_t::edges, extent<1> > weights;
    typedef inout_accessor<2, icosahedral_topology_t::cells> out_cells;
    typedef boost::mpl::vector<rho, weights, out_cells, cell_area> arg_list
        ;

    template<typename Evaluation>
    GT_FUNCTION static void Do(Evaluation const &eval, x_interval)
```

```

{
    auto ff = [](const double _in1, const double _in2, const double
        _res) -> double { return _in1 * _in2 + _res; };

    eval(out_cells()) = eval(on_edges(ff, 0.0, eval(weights), rho()));
}
};

auto weights = icosahedral_grid.make_storage<icosahedral_topology_t::
    cells, icosahedral_topology_t::edges, double >("weights");

```

Note that here `weights` is a field on both `cells` and `edges`. The idea is that by calling `eval(weights)` we get an edge storage regarding current cell. This storage is virtual in the sense that only edges on that cell is stored, other entries are all empty.

References

- [1] Wan, Hui. *Developing and testing a hydrostatic atmospheric dynamical core on triangular grids*. Diss. University of Hamburg Hamburg, 2009.