V-1.0 2025-01-01

Version: 1.0 Updated: 2025-01-01

Section 1

Introduction and Background

1.0.1 Program Workflow

```
egin{align*} 	ext{Starting Point} & 	ext{input:} \ oldsymbol{f} = \Phi^n egin{align*} oldsymbol{uf} = u_f^{n+rac{1}{2}} \ oldsymbol{flux}(	ext{empty}) oldsymbol{dt} = \Delta t \ oldsymbol{src} = \mathbf{g}^n \ 	ext{gradient:} \ oldsymbol{g} = 
abla oldsymbol{f} = 
abla \Phi \ \end{pmatrix}
```

Fluxes Compute

Traversal each elements in tracers (if tracers is vector, then this step traversal component on every direction)

computation:

$$flux = \Phi_f^{n+\frac{1}{2}} u_f^{n+\frac{1}{2}}$$

```
struct Advection {
  scalar * tracers;
  face vector u;
```

V-1.0 2025-01-01 ii

```
double dt;
  scalar * src; // optional
};
void advection (struct Advection p)
{
  scalar * lsrc = p.src;
  if (!lsrc)
    for (scalar s in p.tracers)
        lsrc = list_append (lsrc, zeroc);
  assert (list_len(p.tracers) == list_len(lsrc));
```