Version: 1.0 *Updated: 2025-01-01*

Section 1

Introduction and Background

1.0.1 Program Workflow

Starting Point
input:
$f = \Phi^n$ $uf = u_f^{n+\frac{1}{2}}$
flux(empty) *dt*=Δt
$src = \mathbf{g}^n$
gradient:
$g = \nabla f = \nabla\Phi$

```
void tracer_fluxes (scalar f,
                    face vector uf,
                    face vector flux,
                    double dt,
                    (const) scalar src)
  vector g[];
  gradients ({f}, {g});
```

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;
  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));
```