

```
def adjoint(model, adj_src, adj_rec,
            space_order=2):
    m, eta = model.m, model.damp
    # Allocate wavefield and auxiliary fields
    v = TimeFunction(name='v', grid=model.grid,
                    time_order=2,
                    space_order=space_order)

    # Derive stencil from symbolic equation
    eqn = m * v.dt2 - v.laplace - eta * v.dt
    stencil = solve(eqn, v.backward)
    update_v = Eq(u.backward, stencil)

    # Receiver injection and adj-source
    interpolation
    src_a = adj_src.inject(field=v.backward,
                          expr=rec * dt**2 / m)
    rec_a = adj_rec.interpolate(expr=v)

    op = Operator([update_v] + src_a + rec_a,
                  subs=model.spacing_map)
```