```
def gradient(model, u, adj_src, 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)
    grad = Function(name='grad', grid=model.grid)

    # 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 gradient update
    src_a = adj_src.inject(field=v.backward,
                           expr=rec * dt**2 / m)
    update_grad = Eq(grad, grad - u * v.dt2)

    op = Operator([update_v] + src_a +
         update_grad,
                     subs=model.spacing)
```