

```
def fwi_gradient(model, op_fwd, op_grad):
    """
    Function to compute a single FWI gradient
    """
    u = TimeFunction(name='u', grid=model.grid,
                     space_order=order)
    grad = Function(name='grad', grid=model.grid)

    for i in nshots:
        # Update source location for each shot
        src.coordinates.data[0. :] =
            source_loc[i]

        # Run forward modelling operator
        op_fwd(u=u, src=src, rec=smooth_d)

        # Compute gradient from data residual and
        # update objective function
        residual = smooth_d.data[:] -
            true_d.data[:]
        objective +=
            .5*np.linalg.norm(residual)**2
        op_grad(rec=residual, u=u, m=model.m,
               grad=grad)

    return objective, grad.data
```