The earthquake source is represented by the point force ***f***, which may be written in terms of a moment tensor ***M*** as (omitting the source time function)

 (1)

The point force ***f*** in 3D:

, (2)

Multiplying equations (2) by the time-independent test functions, we can obtain:

, (3)

where ,  and  are the test functions, respectively.

Applying the identity of the Dirac delta function:

, (4)

If we can find the dimensionless parameters  satisfying:

, (5)

Applying (5) and considering the unit test function, equations (4) can be expressed:

 (6)

Therefore, the source force at each GLL node can be written:



(7)

**In your code:**

do m = 1,NGLLZ

do l = 1,NGLLY

do k = 1,NGLLX

xixd = dble(xix(k,l,m))

xiyd = dble(xiy(k,l,m))

xizd = dble(xiz(k,l,m))

etaxd = dble(etax(k,l,m))

etayd = dble(etay(k,l,m))

etazd = dble(etaz(k,l,m))

gammaxd = dble(gammax(k,l,m))

gammayd = dble(gammay(k,l,m))

gammazd = dble(gammaz(k,l,m))

G11(k,l,m) = Mxx\*xixd+Mxy\*xiyd+Mxz\*xizd

G12(k,l,m) = Mxx\*etaxd+Mxy\*etayd+Mxz\*etazd

G13(k,l,m) = Mxx\*gammaxd+Mxy\*gammayd+Mxz\*gammazd

G21(k,l,m) = Mxy\*xixd+Myy\*xiyd+Myz\*xizd

G22(k,l,m) = Mxy\*etaxd+Myy\*etayd+Myz\*etazd

G23(k,l,m) = Mxy\*gammaxd+Myy\*gammayd+Myz\*gammazd

G31(k,l,m) = Mxz\*xixd+Myz\*xiyd+Mzz\*xizd

G32(k,l,m) = Mxz\*etaxd+Myz\*etayd+Mzz\*etazd

G33(k,l,m) = Mxz\*gammaxd+Myz\*gammayd+Mzz\*gammazd

enddo

enddo

enddo

! compute Lagrange polynomials at the source location

call lagrange\_any(xi\_source,NGLLX,xigll,hxis,hpxis)

call lagrange\_any(eta\_source,NGLLY,yigll,hetas,hpetas)

call lagrange\_any(gamma\_source,NGLLZ,zigll,hgammas,hpgammas)

! calculate source array

do m = 1,NGLLZ

do l = 1,NGLLY

do k = 1,NGLLX

sourcearrayd(:,k,l,m) = ZERO

do iv = 1,NGLLZ

do it = 1,NGLLY

do ir = 1,NGLLX

sourcearrayd(1,k,l,m) = sourcearrayd(1,k,l,m) + hxis(ir)\*hetas(it)\*hgammas(iv) &

\*(G11**(ir,it,iv)**\*hpxis(k)\*hetas(l)\*hgammas(m) &

+G12**(ir,it,iv)**\*hxis(k)\*hpetas(l)\*hgammas(m) &

+G13**(ir,it,iv)**\*hxis(k)\*hetas(l)\*hpgammas(m))

sourcearrayd(2,k,l,m) = sourcearrayd(2,k,l,m) + hxis(ir)\*hetas(it)\*hgammas(iv) &

\*(G21**(ir,it,iv)**\*hpxis(k)\*hetas(l)\*hgammas(m) &

+G22**(ir,it,iv)**\*hxis(k)\*hpetas(l)\*hgammas(m) &

+G23**(ir,it,iv)**\*hxis(k)\*hetas(l)\*hpgammas(m))

sourcearrayd(3,k,l,m) = sourcearrayd(3,k,l,m) + hxis(ir)\*hetas(it)\*hgammas(iv) &

\*(G31**(ir,it,iv)**\*hpxis(k)\*hetas(l)\*hgammas(m) &

+G32**(ir,it,iv)**\*hxis(k)\*hpetas(l)\*hgammas(m) &

+G33**(ir,it,iv)**\*hxis(k)\*hetas(l)\*hpgammas(m))

enddo

enddo

enddo

enddo

enddo

enddo

According to the expressions (7), the subscripts associated with G11, G12, G13, G21, G22, G23, G31, G32, and G33 may be not correct. I think that their subscripts should be (k, l, m).

**The corrected version:**

do m = 1,NGLLZ

do l = 1,NGLLY

do k = 1,NGLLX

xixd = dble(xix(k,l,m))

xiyd = dble(xiy(k,l,m))

xizd = dble(xiz(k,l,m))

etaxd = dble(etax(k,l,m))

etayd = dble(etay(k,l,m))

etazd = dble(etaz(k,l,m))

gammaxd = dble(gammax(k,l,m))

gammayd = dble(gammay(k,l,m))

gammazd = dble(gammaz(k,l,m))

G11(k,l,m) = Mxx\*xixd+Mxy\*xiyd+Mxz\*xizd

G12(k,l,m) = Mxx\*etaxd+Mxy\*etayd+Mxz\*etazd

G13(k,l,m) = Mxx\*gammaxd+Mxy\*gammayd+Mxz\*gammazd

G21(k,l,m) = Mxy\*xixd+Myy\*xiyd+Myz\*xizd

G22(k,l,m) = Mxy\*etaxd+Myy\*etayd+Myz\*etazd

G23(k,l,m) = Mxy\*gammaxd+Myy\*gammayd+Myz\*gammazd

G31(k,l,m) = Mxz\*xixd+Myz\*xiyd+Mzz\*xizd

G32(k,l,m) = Mxz\*etaxd+Myz\*etayd+Mzz\*etazd

G33(k,l,m) = Mxz\*gammaxd+Myz\*gammayd+Mzz\*gammazd

enddo

enddo

enddo

! compute Lagrange polynomials at the source location

call lagrange\_any(xi\_source,NGLLX,xigll,hxis,hpxis)

call lagrange\_any(eta\_source,NGLLY,yigll,hetas,hpetas)

call lagrange\_any(gamma\_source,NGLLZ,zigll,hgammas,hpgammas)

! calculate source array

do m = 1,NGLLZ

do l = 1,NGLLY

do k = 1,NGLLX

sourcearrayd(:,k,l,m) = ZERO

do iv = 1,NGLLZ

do it = 1,NGLLY

do ir = 1,NGLLX

sourcearrayd(1,k,l,m) = sourcearrayd(1,k,l,m) + hxis(ir)\*hetas(it)\*hgammas(iv) &

\*(G11**(k,l,m)**\*hpxis(k)\*hetas(l)\*hgammas(m) &

+G12**(k,l,m)**\*hxis(k)\*hpetas(l)\*hgammas(m) &

+G13**(k,l,m)**\*hxis(k)\*hetas(l)\*hpgammas(m))

sourcearrayd(2,k,l,m) = sourcearrayd(2,k,l,m) + hxis(ir)\*hetas(it)\*hgammas(iv) &

\*(G21**(k,l,m)**\*hpxis(k)\*hetas(l)\*hgammas(m) &

+G22**(k,l,m)**\*hxis(k)\*hpetas(l)\*hgammas(m) &

+G23**(k,l,m)**\*hxis(k)\*hetas(l)\*hpgammas(m))

sourcearrayd(3,k,l,m) = sourcearrayd(3,k,l,m) + hxis(ir)\*hetas(it)\*hgammas(iv) &

\*(G31**(k,l,m)**\*hpxis(k)\*hetas(l)\*hgammas(m) &

+G32**(k,l,m)**\*hxis(k)\*hpetas(l)\*hgammas(m) &

+G33**(k,l,m)**\*hxis(k)\*hetas(l)\*hpgammas(m))

enddo

enddo

enddo

enddo

enddo

enddo

**However, the second loop can be simplified.**

do m = 1,NGLLZ

do l = 1,NGLLY

do k = 1,NGLLX

sourcearrayd(:,k,l,m) = ZERO

dsrc\_dx = (G11**(k,l,m)**\*hpxis(k)\*hetas(l)\*hgammas(m) &

+G12**(k,l,m)**\*hxis(k)\*hpetas(l)\*hgammas(m) &

+G13**(k,l,m)**\*hxis(k)\*hetas(l)\*hpgammas(m))

dsrc\_dy = (G21**(k,l,m)**\*hpxis(k)\*hetas(l)\*hgammas(m) &

+G22**(k,l,m)**\*hxis(k)\*hpetas(l)\*hgammas(m) &

+G23**(k,l,m)**\*hxis(k)\*hetas(l)\*hpgammas(m))

dsrc\_dy = (G31**(k,l,m)**\*hpxis(k)\*hetas(l)\*hgammas(m) &

+G32**(k,l,m)**\*hxis(k)\*hpetas(l)\*hgammas(m) &

+G33**(k,l,m)**\*hxis(k)\*hetas(l)\*hpgammas(m))

do iv = 1,NGLLZ

do it = 1,NGLLY

do ir = 1,NGLLX

sourcearrayd(1,k,l,m) = sourcearrayd(1,k,l,m) + hxis(ir)\*hetas(it)\*hgammas(iv) &

\*dsrc\_dx

sourcearrayd(2,k,l,m) = sourcearrayd(2,k,l,m) + hxis(ir)\*hetas(it)\*hgammas(iv) &

\*dsrc\_dy

sourcearrayd(3,k,l,m) = sourcearrayd(3,k,l,m) + hxis(ir)\*hetas(it)\*hgammas(iv) &

\*dsrc\_dz

enddo

enddo

enddo

enddo

enddo

enddo

**If we apply the identity of the Lagrange function, i.e. sum(hxis(1:NGLLX)) = 1, sum(hetas(1:NGLLX)) = 1, and sum(hgammas(1:NGLLZ)) = 1, the three loops can be merged:**

call lagrange\_any(xi\_source,NGLLX,xigll,hxis,hpxis)

call lagrange\_any(eta\_source,NGLLY,yigll,hetas,hpetas)

call lagrange\_any(gamma\_source,NGLLZ,zigll,hgammas,hpgammas)

sourcearray(:,:,:) = ZERO

do m = 1,NGLLZ

do l = 1,NGLLY

do k = 1,NGLLX

xixd = dble(xix(k,l,m))

xiyd = dble(xiy(k,l,m))

xizd = dble(xiz(k,l,m))

etaxd = dble(etax(k,l,m))

etayd = dble(etay(k,l,m))

etazd = dble(etaz(k,l,m))

gammaxd = dble(gammax(k,l,m))

gammayd = dble(gammay(k,l,m))

gammazd = dble(gammaz(k,l,m))

G11(k,l,m) = Mxx\*xixd+Mxy\*xiyd+Mxz\*xizd

G12(k,l,m) = Mxx\*etaxd+Mxy\*etayd+Mxz\*etazd

G13(k,l,m) = Mxx\*gammaxd+Mxy\*gammayd+Mxz\*gammazd

G21(k,l,m) = Mxy\*xixd+Myy\*xiyd+Myz\*xizd

G22(k,l,m) = Mxy\*etaxd+Myy\*etayd+Myz\*etazd

G23(k,l,m) = Mxy\*gammaxd+Myy\*gammayd+Myz\*gammazd

G31(k,l,m) = Mxz\*xixd+Myz\*xiyd+Mzz\*xizd

G32(k,l,m) = Mxz\*etaxd+Myz\*etayd+Mzz\*etazd

G33(k,l,m) = Mxz\*gammaxd+Myz\*gammayd+Mzz\*gammazd

dsrc\_dx = (G11**(k,l,m)**\*hpxis(k)\*hetas(l)\*hgammas(m) &

+G12**(k,l,m)**\*hxis(k)\*hpetas(l)\*hgammas(m) &

+G13**(k,l,m)**\*hxis(k)\*hetas(l)\*hpgammas(m))

dsrc\_dy = (G21**(k,l,m)**\*hpxis(k)\*hetas(l)\*hgammas(m) &

+G22**(k,l,m)**\*hxis(k)\*hpetas(l)\*hgammas(m) &

+G23**(k,l,m)**\*hxis(k)\*hetas(l)\*hpgammas(m))

dsrc\_dy = (G31**(k,l,m)**\*hpxis(k)\*hetas(l)\*hgammas(m) &

+G32**(k,l,m)**\*hxis(k)\*hpetas(l)\*hgammas(m) &

+G33**(k,l,m)**\*hxis(k)\*hetas(l)\*hpgammas(m))

sourcearrayd(1,k,l,m) = sourcearrayd(1,k,l,m) + dsrc\_dx

sourcearrayd(2,k,l,m) = sourcearrayd(2,k,l,m) + dsrc\_dx

sourcearrayd(3,k,l,m) = sourcearrayd(3,k,l,m) + dsrc\_dz

enddo

enddo

enddo

**Certainly, your code is correct for linear mapping element because G(ir,it,iv) = G(k,l,m) in that case. However, it is not correct for nonlinear mapping element!**

**Now, the code is consistent with equations (7).**