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)

$$\boldsymbol{f} = -\boldsymbol{M} \cdot \nabla \delta(\boldsymbol{x} - \boldsymbol{x}_s) \tag{1}$$

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

$$\begin{cases} f_x = -\big[M_{xx}\delta'(x-x_s)\delta(y-y_s)\delta(z-z_s) + M_{xy}\delta(x-x_s)\delta'(y-y_s)\delta(z-z_s) \\ \qquad\qquad + M_{xz}\delta(x-x_s)\delta(y-y_s)\delta'(z-z_s)\big] \\ f_y = -\big[M_{xy}\delta'(x-x_s)\delta(y-y_s)\delta(z-z_s) + M_{yy}\delta(x-x_s)\delta'(y-y_s)\delta(z-z_s) \\ \qquad\qquad + M_{yz}\delta(x-x_s)\delta(y-y_s)\delta'(z-z_s)\big], \\ f_z = -\big[M_{xz}\delta'(x-x_s)\delta(y-y_s)\delta(z-z_s) + M_{yz}\delta(x-x_s)\delta'(y-y_s)\delta(z-z_s) \\ \qquad\qquad + M_{zz}\delta(x-x_s)\delta(y-y_s)\delta'(z-z_s)\big] \end{cases} \tag{2}$$

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

$$\begin{cases} F_x = -\int \big[M_{xx}\delta'(x-x_s)\delta(y-y_s)\delta(z-z_s) + M_{xy}\delta(x-x_s)\delta'(y-y_s)\delta(z-z_s) \\ \qquad\qquad + M_{xz}\delta(x-x_s)\delta(y-y_s)\delta'(z-z_s)\big]\varphi_x(x,y,z)d\Omega \\ F_y = -\int \big[M_{xy}\delta'(x-x_s)\delta(y-y_s)\delta(z-z_s) + M_{yy}\delta(x-x_s)\delta'(y-y_s)\delta(z-z_s) \\ \qquad\qquad + M_{yz}\delta(x-x_s)\delta(y-y_s)\delta'(z-z_s)\big]\varphi_y(x,y,z)d\Omega \\ F_z = -\int \big[M_{xz}\delta'(x-x_s)\delta(y-y_s)\delta(z-z_s) + M_{yz}\delta(x-x_s)\delta'(y-y_s)\delta(z-z_s) \\ \qquad\qquad + M_{zz}\delta(x-x_s)\delta(y-y_s)\delta'(z-z_s)\big]\varphi_z(x,y,z)d\Omega \end{cases}, \tag{3}$$

where φ_x, φ_y and φ_z are the test functions, respectively.

Applying the identity of the Dirac delta function:

$$\begin{cases} F_x(x_s,y_s,z_s) = -\left(M_{xx}\dfrac{\partial\varphi_x(x_s,y_s,z_s)}{\partial x} + M_{xy}\dfrac{\partial\varphi_x(x_s,y_s,z_s)}{\partial y} + M_{xz}\dfrac{\partial\varphi_x(x_s,y_s,z_s)}{\partial z}\right) \\ F_y(x_s,y_s,z_s) = -\left(M_{xy}\dfrac{\partial\varphi_y(x_s,y_s,z_s)}{\partial x} + M_{yy}\dfrac{\partial\varphi_y(x_s,y_s,z_s)}{\partial y} + M_{yz}\dfrac{\partial\varphi_y(x_s,y_s,z_s)}{\partial z}\right), \\ F_z(x_s,y_s,z_s) = -\left(M_{xz}\dfrac{\partial\varphi_z(x_s,y_s,z_s)}{\partial x} + M_{yz}\dfrac{\partial\varphi_z(x_s,y_s,z_s)}{\partial y} + M_{zz}\dfrac{\partial\varphi_z(x_s,y_s,z_s)}{\partial z}\right) \end{cases} \tag{4}$$

If we can find the dimensionless parameters (ξ_s,η_s,γ_s) satisfying:

$$\begin{cases} x_s = \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\ell_i(\xi_s)\ell_j(\eta_s)\ell_k(\gamma_s)x_{ijk} \\ y_s = \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\ell_i(\xi_s)\ell_j(\eta_s)\ell_k(\gamma_s)y_{ijk}, \\ z_s = \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\ell_i(\xi_s)\ell_j(\eta_s)\ell_k(\gamma_s)z_{ijk} \end{cases} \tag{5}$$

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

$$
\begin{cases}
F_x(\xi_s,\eta_s,\gamma_s) = -\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\Bigg\{M_{xx}\left[\ell'_i(\xi_s)\frac{\partial\xi}{\partial x}\ell_j(\eta_s)\ell_k(\gamma_s)+\ell_i(\xi_s)\ell'_j(\eta_s)\frac{\partial\eta}{\partial x}\ell_k(\gamma_s)+\ell_i(\xi_s)\ell_j(\eta_s)\ell'_k(\gamma_s)\frac{\partial\gamma}{\partial x}\right] \\
\qquad +M_{xy}\left[\ell'_i(\xi_s)\frac{\partial\xi}{\partial y}\ell_j(\eta_s)\ell_k(\gamma_s)+\ell_i(\xi_s)\ell'_j(\eta_s)\frac{\partial\eta}{\partial y}\ell_k(\gamma_s)+\ell_i(\xi_s)\ell_j(\eta_s)\ell'_k(\gamma_s)\frac{\partial\gamma}{\partial y}\right] \\
\qquad +M_{xz}\left[\ell'_i(\xi_s)\frac{\partial\xi}{\partial z}\ell_j(\eta_s)\ell_k(\gamma_s)+\ell_i(\xi_s)\ell'_j(\eta_s)\frac{\partial\eta}{\partial z}\ell_k(\gamma_s)+\ell_i(\xi_s)\ell_j(\eta_s)\ell'_k(\gamma_s)\frac{\partial\gamma}{\partial z}\right]\Bigg\} \\
F_y(\xi_s,\eta_s,\gamma_s) = -\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\Bigg\{M_{xy}\left[\ell'_i(\xi_s)\frac{\partial\xi}{\partial x}\ell_j(\eta_s)\ell_k(\gamma_s)+\ell_i(\xi_s)\ell'_j(\eta_s)\frac{\partial\eta}{\partial x}\ell_k(\gamma_s)+\ell_i(\xi_s)\ell_j(\eta_s)\ell'_k(\gamma_s)\frac{\partial\gamma}{\partial x}\right] \\
\qquad +M_{yy}\left[\ell'_i(\xi_s)\frac{\partial\xi}{\partial y}\ell_j(\eta_s)\ell_k(\gamma_s)+\ell_i(\xi_s)\ell'_j(\eta_s)\frac{\partial\eta}{\partial y}\ell_k(\gamma_s)+\ell_i(\xi_s)\ell_j(\eta_s)\ell'_k(\gamma_s)\frac{\partial\gamma}{\partial y}\right] \\
\qquad +M_{yz}\left[\ell'_i(\xi_s)\frac{\partial\xi}{\partial z}\ell_j(\eta_s)\ell_k(\gamma_s)+\ell_i(\xi_s)\ell'_j(\eta_s)\frac{\partial\eta}{\partial z}\ell_k(\gamma_s)+\ell_i(\xi_s)\ell_j(\eta_s)\ell'_k(\gamma_s)\frac{\partial\gamma}{\partial z}\right]\Bigg\} \\
F_z(\xi_s,\eta_s,\gamma_s) = -\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\Bigg\{M_{xz}\left[\ell'_i(\xi_s)\frac{\partial\xi}{\partial x}\ell_j(\eta_s)\ell_k(\gamma_s)+\ell_i(\xi_s)\ell'_j(\eta_s)\frac{\partial\eta}{\partial x}\ell_k(\gamma_s)+\ell_i(\xi_s)\ell_j(\eta_s)\ell'_k(\gamma_s)\frac{\partial\gamma}{\partial x}\right] \\
\qquad +M_{yz}\left[\ell'_i(\xi_s)\frac{\partial\xi}{\partial y}\ell_j(\eta_s)\ell_k(\gamma_s)+\ell_i(\xi_s)\ell'_j(\eta_s)\frac{\partial\eta}{\partial y}\ell_k(\gamma_s)+\ell_i(\xi_s)\ell_j(\eta_s)\ell'_k(\gamma_s)\frac{\partial\gamma}{\partial y}\right] \\
\qquad +M_{zz}\left[\ell'_i(\xi_s)\frac{\partial\xi}{\partial z}\ell_j(\eta_s)\ell_k(\gamma_s)+\ell_i(\xi_s)\ell'_j(\eta_s)\frac{\partial\eta}{\partial z}\ell_k(\gamma_s)+\ell_i(\xi_s)\ell_j(\eta_s)\ell'_k(\gamma_s)\frac{\partial\gamma}{\partial z}\right]\Bigg\}
\end{cases}
\tag{6}
$$

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

$$
\begin{cases}
F_x(i,j,k) = -\Bigg\{M_{xx}\left[\ell'_i(\xi_s)\frac{\partial\xi}{\partial x}\ell_j(\eta_s)\ell_k(\gamma_s)+\ell_i(\xi_s)\ell'_j(\eta_s)\frac{\partial\eta}{\partial x}\ell_k(\gamma_s)+\ell_i(\xi_s)\ell_j(\eta_s)\ell'_k(\gamma_s)\frac{\partial\gamma}{\partial x}\right] \\
\qquad +M_{xy}\left[\ell'_i(\xi_s)\frac{\partial\xi}{\partial y}\ell_j(\eta_s)\ell_k(\gamma_s)+\ell_i(\xi_s)\ell'_j(\eta_s)\frac{\partial\eta}{\partial y}\ell_k(\gamma_s)+\ell_i(\xi_s)\ell_j(\eta_s)\ell'_k(\gamma_s)\frac{\partial\gamma}{\partial y}\right] \\
\qquad +M_{xz}\left[\ell'_i(\xi_s)\frac{\partial\xi}{\partial z}\ell_j(\eta_s)\ell_k(\gamma_s)+\ell_i(\xi_s)\ell'_j(\eta_s)\frac{\partial\eta}{\partial z}\ell_k(\gamma_s)+\ell_i(\xi_s)\ell_j(\eta_s)\ell'_k(\gamma_s)\frac{\partial\gamma}{\partial z}\right]\Bigg\} \\
F_y(i,j,k) = -\Bigg\{M_{xy}\left[\ell'_i(\xi_s)\frac{\partial\xi}{\partial x}\ell_j(\eta_s)\ell_k(\gamma_s)+\ell_i(\xi_s)\ell'_j(\eta_s)\frac{\partial\eta}{\partial x}\ell_k(\gamma_s)+\ell_i(\xi_s)\ell_j(\eta_s)\ell'_k(\gamma_s)\frac{\partial\gamma}{\partial x}\right] \\
\qquad +M_{yy}\left[\ell'_i(\xi_s)\frac{\partial\xi}{\partial y}\ell_j(\eta_s)\ell_k(\gamma_s)+\ell_i(\xi_s)\ell'_j(\eta_s)\frac{\partial\eta}{\partial y}\ell_k(\gamma_s)+\ell_i(\xi_s)\ell_j(\eta_s)\ell'_k(\gamma_s)\frac{\partial\gamma}{\partial y}\right] \\
\qquad +M_{yz}\left[\ell'_i(\xi_s)\frac{\partial\xi}{\partial z}\ell_j(\eta_s)\ell_k(\gamma_s)+\ell_i(\xi_s)\ell'_j(\eta_s)\frac{\partial\eta}{\partial z}\ell_k(\gamma_s)+\ell_i(\xi_s)\ell_j(\eta_s)\ell'_k(\gamma_s)\frac{\partial\gamma}{\partial z}\right]\Bigg\} \\
F_z(i,j,k) = -\Bigg\{M_{xz}\left[\ell'_i(\xi_s)\frac{\partial\xi}{\partial x}\ell_j(\eta_s)\ell_k(\gamma_s)+\ell_i(\xi_s)\ell'_j(\eta_s)\frac{\partial\eta}{\partial x}\ell_k(\gamma_s)+\ell_i(\xi_s)\ell_j(\eta_s)\ell'_k(\gamma_s)\frac{\partial\gamma}{\partial x}\right] \\
\qquad +M_{yz}\left[\ell'_i(\xi_s)\frac{\partial\xi}{\partial y}\ell_j(\eta_s)\ell_k(\gamma_s)+\ell_i(\xi_s)\ell'_j(\eta_s)\frac{\partial\eta}{\partial y}\ell_k(\gamma_s)+\ell_i(\xi_s)\ell_j(\eta_s)\ell'_k(\gamma_s)\frac{\partial\gamma}{\partial y}\right] \\
\qquad +M_{zz}\left[\ell'_i(\xi_s)\frac{\partial\xi}{\partial z}\ell_j(\eta_s)\ell_k(\gamma_s)+\ell_i(\xi_s)\ell'_j(\eta_s)\frac{\partial\eta}{\partial z}\ell_k(\gamma_s)+\ell_i(\xi_s)\ell_j(\eta_s)\ell'_k(\gamma_s)\frac{\partial\gamma}{\partial z}\right]\Bigg\}
\end{cases}
\tag{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).