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

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

The point force $\boldsymbol{f}$ in 2D:

$$\begin{cases} f_x = -[M_{xx}\delta'(x-x_s)\delta(z-z_s) + M_{xz}\delta(x-x_s)\delta'(z-z_s)] \\ f_z = -[M_{xz}\delta'(x-x_s)\delta(z-z_s) + M_{zz}\delta(x-x_s)\delta'(z-z_s)] \end{cases}, \tag{2}$$

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

$$\begin{cases} F_x(x,z) = -\int [M_{xx}\delta'(x-x_s)\delta(z-z_s) + M_{xz}\delta(x-x_s)\delta'(z-z_s)]\varphi_x(x,z)d\Omega \\ F_z(x,z) = -\int [M_{xz}\delta'(x-x_s)\delta(z-z_s) + M_{zz}\delta(x-x_s)\delta'(z-z_s)]\varphi_z(x,z)d\Omega \end{cases}, \tag{3}$$

where φ_x and φ_z are the test functions, respectively.

Applying the identity of the Dirac delta function:

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

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

$$\begin{cases} x_s = \displaystyle\sum_{i=1}^{N}\sum_{j=1}^{N} \ell_i(\xi_s)\ell_j(\gamma_s)x_{ij} \\ z_s = \displaystyle\sum_{i=1}^{N}\sum_{j=1}^{N} \ell_i(\xi_s)\ell_j(\gamma_s)z_{ij} \end{cases}, \tag{5}$$

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

$$\begin{aligned} F_x(\xi_s,\gamma_s) = -\sum_{i=1}^{N}\sum_{j=1}^{N} \Bigg\{ & M_{xx}\left[\ell_i'(\xi_s)\frac{\partial \xi}{\partial x}\ell_j(\gamma_s) + \ell_i(\xi_s)\ell_j'(\gamma_s)\frac{\partial \gamma}{\partial x} \right] \\ & + M_{xz}\left[\ell_i'(\xi_s)\frac{\partial \xi}{\partial z}\ell_j(\gamma_s) + \ell_i(\xi_s)\ell_j'(\gamma_s)\frac{\partial \gamma}{\partial z} \right] \Bigg\} \\ F_z(\xi_s,\gamma_s) = -\sum_{i=1}^{N}\sum_{j=1}^{N} \Bigg\{ & M_{xz}\left[\ell_i'(\xi_s)\frac{\partial \xi}{\partial x}\ell_j(\gamma_s) + \ell_i(\xi_s)\ell_j'(\gamma_s)\frac{\partial \gamma}{\partial x} \right] \\ & + M_{zz}\left[\ell_i'(\xi_s)\frac{\partial \xi}{\partial z}\ell_j(\gamma_s) + \ell_i(\xi_s)\ell_j'(\gamma_s)\frac{\partial \gamma}{\partial z} \right] \Bigg\} \end{aligned}, \tag{6}$$

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

$$
\begin{aligned}
F_x(i,j) = -\Big\{ & M_{xx}\left[\ell'_i(\xi_s)\frac{\partial\xi}{\partial x}\ell_j(\gamma_s)+\ell_i(\xi_s)\ell'_j(\gamma_s)\frac{\partial\gamma}{\partial x}\right] \\
& + M_{xz}\left[\ell'_i(\xi_s)\frac{\partial\xi}{\partial z}\ell_j(\gamma_s)+\ell_i(\xi_s)\ell'_j(\gamma_s)\frac{\partial\gamma}{\partial z}\right]\Big\} \\
F_z(i,j) = -\Big\{ & M_{xz}\left[\ell'_i(\xi_s)\frac{\partial\xi}{\partial x}\ell_j(\gamma_s)+\ell_i(\xi_s)\ell'_j(\gamma_s)\frac{\partial\gamma}{\partial x}\right] \\
& + M_{zz}\left[\ell'_i(\xi_s)\frac{\partial\xi}{\partial z}\ell_j(\gamma_s)+\ell_i(\xi_s)\ell'_j(\gamma_s)\frac{\partial\gamma}{\partial z}\right]\Big\}
\end{aligned}
\tag{7}
$$

In your code:

```
do m=1,NGLLZ
   do k=1,NGLLX

      xixd     = xix(k,m,ispec_selected_source)
      xizd     = xiz(k,m,ispec_selected_source)
      gammaxd = gammax(k,m,ispec_selected_source)
      gammazd = gammaz(k,m,ispec_selected_source)

      G11(k,m) = Mxx*xixd+Mxz*xizd
      G13(k,m) = Mxx*gammaxd+Mxz*gammazd
      G31(k,m) = Mxz*xixd+Mzz*xizd
      G33(k,m) = Mxz*gammaxd+Mzz*gammazd

   end do
end do

! compute Lagrange polynomials at the source location
call lagrange_any(xi_source,NGLLX,xigll,hxis,hpxis)
call lagrange_any(gamma_source,NGLLZ,zigll,hgammas,hpgammas)

! calculate source array
```

```
do m=1,NGLLZ
   do k=1,NGLLX

      sourcearray(:,k,m) = ZERO

      do iv=1,NGLLZ
         do ir=1,NGLLX

            sourcearray(1,k,m) = sourcearray(1,k,m) + hxis(ir)*hgammas(iv) &
                                          *(G11(ir,iv)*hpxis(k)*hgammas(m) &
                                          +G13(ir,iv)*hxis(k)*hpgammas(m))


            sourcearray(2,k,m) = sourcearray(2,k,m) + hxis(ir)*hgammas(iv) &
                                         *(G31(ir,iv)*hpxis(k)*hgammas(m) &
                                         +G33(ir,iv)*hxis(k)*hpgammas(m))

         end do
      end do

   end do
end do
```

According to the expression (7), the subscripts associated with G11, G13, G31 and G33 may be not correct. I think that their subscripts should be (k, m).

The corrected version:

```
do m=1,NGLLZ
    do k=1,NGLLX
```

```
          xixd       = xix(k,m,ispec_selected_source)
          xizd       = xiz(k,m,ispec_selected_source)
          gammaxd = gammax(k,m,ispec_selected_source)
          gammazd = gammaz(k,m,ispec_selected_source)

          G11(k,m) = Mxx*xixd+Mxz*xizd
          G13(k,m) = Mxx*gammaxd+Mxz*gammazd
          G31(k,m) = Mxz*xixd+Mzz*xizd
          G33(k,m) = Mxz*gammaxd+Mzz*gammazd

      end do
end do

! compute Lagrange polynomials at the source location
call lagrange_any(xi_source,NGLLX,xigll,hxis,hpxis)
call lagrange_any(gamma_source,NGLLZ,zigll,hgammas,hpgammas)

! calculate source array
do m=1,NGLLZ
   do k=1,NGLLX

          sourcearray(:,k,m) = ZERO

          do iv=1,NGLLZ
             do ir=1,NGLLX

                sourcearray(1,k,m) = sourcearray(1,k,m) + hxis(ir)*hgammas(iv) &
                                                    *(G11(k,m)*hpxis(k)*hgammas(m) &
                                                    +G13(k,m)*hxis(k)*hpgammas(m))
```

```
          sourcearray(2,k,m) = sourcearray(2,k,m) + hxis(ir)*hgammas(iv) &
                                              *(G31(k,m)*hpxis(k)*hgammas(m) &
                                              +G33(k,m)*hxis(k)*hpgammas(m))

        end do
      end do

   end do
end do
```

However, the second loop can be simplified.

```
do m=1,NGLLZ
   do k=1,NGLLX

      sourcearray(:,k,m) = ZERO

      dsrc_dx = (G11(k,m)*hpxis(k)*hgammas(m) &
                +G13(k,m)*hxis(k)*hpgammas(m))
      dsrc_dz = (G31(k,m)*hpxis(k)*hgammas(m) &
                +G33(k,m)*hxis(k)*hpgammas(m))
      do iv=1,NGLLZ
        do ir=1,NGLLX

          sourcearray(1,k,m) = sourcearray(1,k,m) + hxis(ir)*hgammas(iv) &
                                               *dsrc_dx

          sourcearray(2,k,m) = sourcearray(2,k,m) + hxis(ir)*hgammas(iv) &
                                               *dsrc_dz
```

```
        end do
      end do

   end do
end do
```

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

```
call lagrange_any(xi_source,NGLLX,xigll,hxis,hpxis)
call lagrange_any(gamma_source,NGLLZ,zigll,hgammas,hpgammas)

sourcearray(:,:,:) = ZERO

do m=1,NGLLZ
   do k=1,NGLLX

      xixd     = xix(k,m,ispec_selected_source)
      xizd     = xiz(k,m,ispec_selected_source)
      gammaxd = gammax(k,m,ispec_selected_source)
      gammazd = gammaz(k,m,ispec_selected_source)

      G11(k,m) = Mxx*xixd+Mxz*xizd
      G13(k,m) = Mxx*gammaxd+Mxz*gammazd
      G31(k,m) = Mxz*xixd+Mzz*xizd
      G33(k,m) = Mxz*gammaxd+Mzz*gammazd

      dsrc_dx = (G11(k,m)*hpxis(k)*hgammas(m) &
                +G13(k,m)*hxis(k)*hpgammas(m))
```

```
        dsrc_dz = (G31(k,m)*hpxis(k)*hgammas(m) &
                    +G33(k,m)*hxis(k)*hpgammas(m))

        sourcearray(1,k,m) = sourcearray(1,k,m) + dsrc_dx
        sourcearray(2,k,m) = sourcearray(2,k,m) + dsrc_dz

    end do
end do
```

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

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