```
subroutine theta_geom2theta_flux(s,theta_geom_vec,theta_flux_vec)
      implicit none
      double precision :: s
      double precision, dimension(:), intent(in) :: theta_geom_vec
      double precision, dimension(:), intent(out) :: theta_flux_vec
      integer :: ntheta, i, i_theta_geom
      integer, parameter :: ntheta_interp = 500, nplag = 10
      double precision, dimension(nplag) :: coef
      double precision :: psi,theta,q,dq_ds,sqrtg,bmod,dbmod_dtheta,R,dR_ds,dR_dtheta,Z,dZ_ds,dZ_dtheta,R0,Z0
      double precision, dimension(ntheta_interp+nplag) :: theta_geom_interp, theta_flux_interp
      ntheta = size(theta_geom_vec)
      theta_flux_interp = dfloat([(i, i=-nplag/2,ntheta_interp-1+nplag/2, 1)])/ntheta_interp* 2.d0*pi
      !Calculate position of magnetic axis
      call magdata_in_symfluxcoord_ext(1,0.d0,psi,0.d0,q,dq_ds, &
                                       & sqrtg,bmod,dbmod_dtheta,R0,dR_ds,dR_dtheta, &
                                       & Z0,dZ_ds,dZ_dtheta)
      !Calculate sampling points for interpolation
      do i = 1,ntheta_interp+nplag !size theta_flux_interp
         theta = modulo(theta_flux_interp(i),2.d0*pi)
         call magdata_in_symfluxcoord_ext(1, s,psi,theta,q,dq_ds,sqrtg,bmod,dbmod_dtheta, &
                                          R,dR_ds,dR_dtheta,Z,dZ_ds,dZ_dtheta)
         theta_geom_interp(i) = modulo(atan2((Z-Z0),(R-R0)),2.d0*pi)-2.d0*pi !to shift it in the same regime
         if ((i.gt.1).and.(theta_geom_interp(i).lt.theta_geom_interp(i-1))) then
            !Extend the coordinate regime over 2*pi, to guarantee smoothness of the curve
            theta_geom_interp(i) = theta_geom_interp(i)+
2.d0*pi*(ceiling((theta_geom_interp(i-1)-(theta_geom_interp(i)))/(2.d0*pi)))
         endif
      enddo
      !Interpolate with Lagrange polynomials
      do i = 1,ntheta
         if (theta_geom_vec(i) .eq. 0.d0) then
            theta_flux_vec(i) = 0.d0
            cycle
         endif
         !Find indices of theta_geom_vec(i) among sampling points
         call binsrc(theta_geom_interp(nplag/2+1:ntheta_interp+nplag/2),1,ntheta_interp,theta_geom_vec(i),i_theta_geom)
         i_theta_geom = i_theta_geom + nplag/2
      !
         call plag_coeff(nplag,0,theta_geom_vec(i),theta_geom_interp(i_theta_geom-nplag/2:i_theta_geom+nplag/2-1),coef)
         theta_flux_vec(i) = modulo(sum(coef*theta_flux_interp(i_theta_geom-nplag/2:i_theta_geom+nplag/2-1)),2.d0*pi)
      enddo
end subroutine theta_geom2theta_flux
```