```
function [B,det_J] = ElementTransformation(Eval_DShapeFn,GridPts,choice)
% $Author : Vignesh Ramakrishnan$
% $RIN : 662028006$  $Date : November 24, 2021$
% $Code Version: 1.0$
% This function evaluates the transformation of evaluated local Shape
% Function Gradients from local coordinate system to global coordinate
% system and also compute the determinant of jacobian at the integration
% points.
% Inputs : Eval_DShapeFn - Gradient of Shape functions evaluated at
%                          Quadrature or integration points
%          GridPts       - global GridPts of nodes of the elements
%          choice        - choice of integrator.
% Outputs: B             - Transformed Gradient of Shape functions at
%                          Quadrature points
%          det_J         - determinant of Jacobians at global nodal
%                          locations of nodes of the element

    [dim,n,num_IntPts] = size(Eval_DShapeFn);
    B = zeros(dim,n,num_IntPts);
    det_J = zeros(1,num_IntPts);
    det_tol = 1e-4;

    if choice == 3 % diffusion
        for i=1:num_IntPts
            t = Eval_DShapeFn(:,:,i);
            J = t*GridPts;
            J = J';
            det_J(1,i) = det(J);
            if det_J(1,i) <= det_tol
                det_J(1,i) = det_tol;
            end
            cofJ = (adjoint(J));
            B(:,:,i) = (1/det_J(1,i))*cofJ*Eval_DShapeFn(:,:,i);
        end
    end

    if (choice == 2 || choice == 1) % convection - 2, mass integrator - 1
        for i=1:num_IntPts
            t = Eval_DShapeFn(:,:,i);
            J = t*GridPts;
            J = J';
            det_J(1,i) = det(J);
            if det_J(1,i) <= det_tol
                det_J(1,i) = det_tol;
            end
            cofJ = (adjoint(J));
            B(:,:,i) = (1/det_J(1,i))*cofJ*Eval_DShapeFn(:,:,i);
        end
    end

end
```

Published with MATLAB® R2021a