```
function [localmat] = inner_prod_ten0(triag_no, quad_rul, scal_fun,
ten0_type)

%
% This function computes, the values for integral of f*v
% at the requested xy_pts points in triangle triag_no.
%  The vector of values is returned in localmat.
%
%

%%%%%%%%%%%%%%%%%%%%%% Global Variables %%%%%%%%%%%%%%%%%%%
global nodeco  elnode  bdynde  bdyedge  nVert  nedge
global GlobalV  GlobalP  GlobalS  GlobalG
global dimTvel  dimTpre  dimTstr  dimTGrv
global vel_bas_type  pre_bas_type  str_bas_type  Grv_bas_type
global quad_rul num

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Description of triangle.
cotri(1:3,1) = nodeco(elnode(triag_no, 1:3), 1) ;
cotri(1:3,2) = nodeco(elnode(triag_no, 1:3), 2) ;

Jmat = [(cotri(2,1) - cotri(1,1)), (cotri(3,1) - cotri(1,1)) ; ...
        (cotri(2,2) - cotri(1,2)) , (cotri(3,2) - cotri(1,2)) ] ;
detJ = abs(Jmat(1,1)*Jmat(2,2) - Jmat(1,2)*Jmat(2,1));
JInv = inv(Jmat) ;

% Evaluation of quadrature points and quadrature weights.
[quad_pts, quad_wghts] = feval(quad_rul) ;
nqpts = size(quad_pts,1) ;

% Adjust points and weights to account for size of true triangle.
xy_pts = ( Jmat * quad_pts.' ).' ;
xy_pts(:,1) = cotri(1,1) + xy_pts(:,1) ;
xy_pts(:,2) = cotri(1,2) + xy_pts(:,2) ;
quad_wghts = detJ * quad_wghts ;

% Evaluate the scalar multiplier at the quadrature points.
sfun_vals = feval(scal_fun, xy_pts, triag_no) ;

% Evaluate Basis Functions and their Gradients at quad. points.
[ten0a, Gradten0a] = feval(ten0_type, quad_pts) ;
nbas0a = size(ten0a,1) ;

% Now to do the evaluations of the integrals.
for iq = 1:nqpts
   ten0a(:,iq) = quad_wghts(iq) * ten0a(:,iq) ;
end

mat1 = ten0a*sfun_vals';
```

```
localmat = [ mat1 ] ;
```

Published with MATLAB® R2023b